

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка програмного забезпечення підтримки процесів автоматизованого управління та аналізу витрат в домогосподарствах мовою Python»

на здобуття освітнього ступеня бакалавра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Олексій КАЗЬМІРЧУК
(підпис)

Виконав: здобувач вищої освіти групи ПД-43

_____ Олексій КАЗЬМІРЧУК

Керівник: _____ Володимир САДОВЕНКО
к.ф.-м.н., доцент

Рецензент: _____

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« ____ » _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Казьмірчуку Олексію Геннадійовичу _____

1. Тема кваліфікаційної роботи: «Розробка програмного забезпечення підтримки процесів автоматизованого управління та аналізу витрат в домогосподарствах мовою Python»

керівник кваліфікаційної роботи к.ф-м.н., доцент Володимир САДОВЕНКО, затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024 р. № 36.

2. Строк подання кваліфікаційної роботи «28» травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: теоретичні відомості про процеси управління та аналізу витрат в домогосподарствах, опис методів управління та аналізу витрат в домогосподарствах, опис існуючих засобів для управління та аналізу витрат в домогосподарствах, технічна документація з описом засобів реалізації автоматизації процесу управління та аналізу витрат.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Дослідження теоретичних основ процесу управління та аналізу витрат в домогосподарствах.

2. Огляд існуючих засобів для управління та аналізу витрат в домогосподарствах.
3. Постановка технічного завдання та вибір засобів реалізації програмного забезпечення для автоматизованого управління та аналізу витрат в домогосподарствах.
4. Розробка та тестування додатку для автоматизованого управління та аналізу витрат в домогосподарствах.
5. Перелік графічного матеріалу: *презентація*
 1. Аналіз аналогів.
 2. Вимоги до додатку.
 3. Програмні засоби реалізації.
 4. Діаграма варіантів використання.
 5. Діаграма діяльності.
 6. Діаграма класів.
 7. Схема баз даних.
 8. Мапа сайту.
 9. Екранні форми.
 10. Апробація результатів дослідження
6. Дата видачі завдання «28» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	28.02.-06.03.2024	
2	Аналіз та дослідження існуючих аналогів	07.03.-13.03.2024	
3	Дослідження теоретичних основ управління та аналізу витрат у домогосподарствах	14.03.-18.03.2024	
4	Постановка технічного завдання для розробки програмного забезпечення	19.03.-25.03.2024	
5	Вибір засобів реалізації додатку	26.03.-02.04.2024	
6	Розробка та тестування додатку для автоматизованого управління та аналізу витрат у домогосподарствах	03.04.-28.04.2024	
7	Оформлення роботи: вступ, висновки, реферат	29.04-05.05.2024	
8	Розробка демонстраційних матеріалів	06.05-12.05.2024	
9	Попередній захист роботи	13.05-31.05.2024	

Здобувач вищої освіти

(підпис)

Олексій КАЗЬМІРЧУК

Керівник
кваліфікаційної роботи

(підпис)

Володимир САДОВЕНКО

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 57 стор., 1 табл., 22 рис., 17 джерел.

Мета роботи – спрощення управління та аналізу витрат в домогосподарствах.

Об'єкт дослідження – процес автоматизованого управління та аналізу витрат в домогосподарствах.

Предмет дослідження – програмне забезпечення, яке допомагає автоматизувати та спростити процеси управління фінансами в домогосподарствах.

Короткий зміст роботи: В роботі досліджено методи автоматизації управління та аналізу витрат в домогосподарствах. Проведено огляд існуючих програмних засобів для управління фінансами, таких як YNAB, Personal Capital, RocketGuard. Розроблено архітектуру програмного забезпечення та програмно реалізовані ключові функціональні можливості додатку, зокрема: реєстрація користувачів, додавання та категоризація витрат, створення та моніторинг загального бюджету та бюджетів за категоріями, візуалізація витрат у вигляді графіків. Проведено мануальне тестування додатку. В роботі використано технології Python для бекенду, JavaScript для фронтенду, а також бібліотеки для роботи з базами даних та візуалізації даних.

Сферою використання застосунку є оптимізація фінансового планування та контролю витрат для індивідуальних користувачів, сімей та малого бізнесу, що сприяє підвищенню фінансової грамотності та стабільності.

КЛЮЧОВІ СЛОВА: УПРАВЛІННЯ ВИТРАТАМИ, АНАЛІЗ ВИТРАТ, БЮДЖЕТ, ДОМОГОСПОДАРСТВО, АВТОМАТИЗОВАНЕ УПРАВЛІННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ УПРАВЛІННЯ ТА АНАЛІЗУ ВИТРАТ

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ.....	8
ВСТУП.....	9
1 ТЕОРЕТИЧНІ СНОВИ УПРАВЛІННЯ ТА АНАЛІЗУ	11
1.1 Домогосподарства та їх роль у економіці держави.....	11
1.2 Витрати, їх види	12
1.3 Соціально-економічна сутність процесу управління та аналізу витрат, ефективність їх застосування для домогосподарств	15
1.4 Методи та принципи управління та аналізу витрат в домогосподарствах	17
2 ДОСЛІДЖЕННЯ ЗАСОБІВ ДЛЯ УПРАВЛІННЯ ТА АНАЛІЗУ ВИТРАТ У ДОМОГОСПОДАРСТВАХ.....	20
2.1 Аналіз інструментальних засобів для управління та аналізу витрат в домогосподарствах.....	20
2.2 Автоматизоване управління та аналіз витрат в домогосподарствах як засіб для підвищення ефективності їх господарювання, його переваги	28
ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	32
3.1 Визначення вимог до програмного забезпечення	32
3.2 Засоби реалізації програмного забезпечення для автоматизованого управління та аналізу витрат в домогосподарствах	33
ДЛЯ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ТА АНАЛІЗУ ВИТРАТ.....	45
4.1 Опис етапів розробки.....	45
4.2 Моделювання додатку	46
4.3 Діаграма варіантів використання	47
4.4 Проектування інтерфейсу	49
4.5 Діаграма діяльності.....	54
4.6 Діаграма класів	55
4.7 Схема баз даних.....	58
4.8 Мапа сайту	60
4.9 Тестування програмного забезпечення.....	62
ВИСНОВКИ.....	64
ПЕРЕЛІК ПОСИЛАНЬ	66
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	68

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ

SQL	-	Structured query language
HTML	-	HyperText Markup Language
CSS	-	Cascading Style Sheets
JS	-	JavaScript
HTTP	-	REpresentational State Transfer
ACID	-	Atomicity, Consistency, Isolation, Durability
DOM	-	Document Object Model
ORM	-	Object-Relational Mapping
UI	-	User Interface
CSV	-	Comma-Separated Values
UML	-	Unified Modeling Language

ВСТУП

Актуальність теми. Розробка програмного забезпечення для підтримки процесів автоматизованого управління та аналізу витрат в домогосподарствах є важливою в сучасному світі, що обумовлюється низкою факторів:

- ускладнення економічних умов, зростання кількості фінансових продуктів та послуг, необхідність прийняття обґрунтованих рішень щодо управління особистими фінансами, призводить до зростання потреби у фінансовій грамотності та безпеці домогосподарств;

- труднощі та час, що витрачається на ведення обліку витрат, складність аналізу даних про витрати, неможливість прогнозувати майбутні витрати, є недоліками традиційних методів управління та аналізу витрат.

Переваги розробки програмного забезпечення для підтримки процесів автоматизованого управління та аналізу витрат в домогосподарствах:

- ефективність – програмне забезпечення допоможе користувачам економити час та гроші;

- простота використання – програмне забезпечення розроблятиметься з урахуванням потреб користувачів;

- функціональність – програмне забезпечення виконуватиме широкий спектр функцій, які допоможуть користувачам управляти своїми фінансами.

Розробка програмного забезпечення для підтримки процесів автоматизованого управління та аналізу витрат в домогосподарствах має значний потенціал для підвищення фінансової грамотності, поліпшення якості життя людей, стимулювання економічного розвитку.

Об'єкт дослідження – процес управління та аналізу витрат в домогосподарствах.

Предмет дослідження – програмне забезпечення, яке допомагає автоматизувати процеси управління та аналізу витрат в домогосподарствах.

Мета дослідження – спрощення управління та аналізу витрат в домогосподарствах за допомогою застосування мовою Python.

Для досягнення мети дослідження визначені основні задачі:

1. Проаналізувати управління та аналіз витрат в домогосподарствах.
2. Проаналізувати засоби, які можуть бути використані для управління та аналізу витрат в домогосподарствах.
3. Визначити переваги та недоліки існуючих програмних засобів.
4. Розробити функціональні та нефункціональні вимоги до застосунку для автоматизованого управління та аналізу витрат в домогосподарствах.
5. Спроекувати архітектуру, визначити класи для створення застосунку для автоматизованого управління та аналізу витрат в домогосподарствах.
6. Розробити програмне забезпечення для автоматизованого управління та аналізу витрат в домогосподарствах.
7. Провести тестування застосунку для автоматизованого управління та аналізу витрат в домогосподарствах.

Завдання дослідження спрямовані на створення програмного забезпечення, яке допоможе домогосподарствам ефективно управляти своїми фінансами, аналізувати витрати та планувати свої фінансові ресурси, зменшити витрати часу для вирішення задач управління та аналізу витрат.

Для досягнення мети дослідження та вирішення поставлених завдань, використаний комплекс методів наукового пізнання: теоретичні (аналіз наукової літератури, інструментальних засобів, програмного забезпечення, моделювання, порівняння тощо), емпіричні (програмування, тестування тощо), математично-статистичні.

Усі ці методи можуть використовуватися як окремо, так і в поєднанні з іншими для досягнення максимальної ефективності в дослідженні та розробці програмного забезпечення.

Практичне значення автоматизації управління та аналізу витрат в домогосподарствах полягає в тому, що даний продукт дозволить користувачам ефективніше контролювати свої фінанси, знижувати витрати та досягати фінансових цілей.

1 ТЕОРЕТИЧНІ СНОВИ УПРАВЛІННЯ ТА АНАЛІЗУ ВИТРАТ У ДОМОГОСПОДРСТВАХ

1.1 Домогосподарства та їх роль у економіці держави

Домогосподарства – це соціально-економічна одиниця, яка складається з однієї або більше осіб, які спільно проживають, ведуть спільне господарство та мають спільний бюджет, незалежно від родинних зв'язків між ними.

Домогосподарство може включати членів однієї сім'ї або декількох сімей, а також осіб, які не мають родинних зв'язків, але проживають разом і спільно управляють своїми ресурсами та витратами. Основними характеристиками домогосподарства є спільне використання житла, ведення спільного бюджету, спільна участь у прийнятті рішень щодо витрат і управління ресурсами. Домогосподарство може складатися як з однієї особи, так і з кількох людей, які разом планують і розподіляють свої фінансові ресурси для забезпечення життєвих потреб і досягнення спільних цілей.

Домогосподарства відіграють важливу роль у економіці, оскільки вони є основними споживачами товарів і послуг, а також постачальниками робочої сили. Їхні рішення щодо витрат та заощаджень впливають на загальний економічний розвиток та рівень життя в суспільстві.

Основними функціями домогосподарств є споживання, заощадження та інвестування. Вони приймають рішення щодо розподілу доходів між поточним споживанням, такими як харчування, одяг, житло, та заощадженнями, які можуть бути використані для майбутніх потреб або інвестування в різні активи. Домогосподарства також можуть брати участь у виробничій діяльності, наприклад, шляхом ведення малого бізнесу або надання послуг.

Взагалі, домогосподарства, як фінансово-економічні структури, мають свої особливості, які визначають їхню роль у економіці та взаємовідносини з іншими суб'єктами економіки.

Перш за все, домогосподарства отримують доходи з різних джерел, таких як заробітна плата, пенсії, інвестиційні доходи, соціальні виплати тощо. Ця різноманітність створює можливість розподілу доходів на різні цілі та споживчі потреби. Вони є основними споживачами товарів і послуг. Їхні рішення щодо витрат визначають загальний попит на товари та послуги, вносять вагомий внесок в економіку держави.

Також, домогосподарства можуть здійснювати збереження та накопичення ресурсів на майбутнє, наприклад, через заощадження в банках або інвестування в цінні папери.

Домогосподарства піддаються різноманітним фінансовим ризикам, таким як втрата робочого місця, медичні витрати або природні катастрофи. Вони можуть здійснювати заходи щодо мінімізації цих ризиків через страхування та накопичення резервів, що також має великий вплив на економіку.

Домогосподарства можуть прагнути досягнути фінансової незалежності, щоб мати достатньо ресурсів для забезпечення своїх потреб і мети без постійної залежності від зовнішніх джерел фінансування. Кожне домогосподарство має свої унікальні фінансові цілі, такі як придбання житла, оплата освіти дітей, відпочинок тощо. Ці цілі визначають стратегії фінансового планування.

Для того, щоб керувати своїми доходами та витратами, планувати майбутні витрати та досягати фінансових цілей домогосподарства повинні проводити бюджетування та фінансове планування.

Для мінімізації допущених помилок, втрат та неефективного використання ресурсів домогосподарствам потрібна допомога. Такою допомогою може стати програмне забезпечення для автоматизованого управління та аналізу витрат в домогосподарствах.

1.2 Витрати, їх види

Витрати, як економічна категорія, є одним із ключових елементів економічної теорії та практики, що характеризують процес виробництва, розподілу

та споживання ресурсів у суспільстві. Вони відображають використання ресурсів для досягнення певних цілей і можуть бути визначені як сукупність всіх витрат, що виникають у процесі виробництва товарів або надання послуг.

Взагалі, витрати – це грошові чи матеріальні ресурси, які витрачаються на придбання товарів або послуг, або на інші цілі, що призводить до зменшення активів чи збільшення зобов'язань.

Витрати виробництва включають всі витрати, пов'язані з процесом створення товарів або послуг. Це може бути вартість сировини, матеріалів, енергії, амортизації обладнання, заробітної плати працівників та інших витрат, необхідних для функціонування виробничого процесу. Витрати виробництва є важливим показником, оскільки вони впливають на собівартість продукції та її ринкову ціну.

Витрати поділяються на прямі та непрямі. Прямі витрати – це витрати, які безпосередньо пов'язані з виробництвом певного продукту або послуги, такі як вартість сировини або заробітна плата працівників, зайнятих у виробництві. Непрямі витрати – це витрати, які не можуть бути безпосередньо віднесені до конкретного продукту або послуги, наприклад, витрати на утримання адміністративного апарату або оренда приміщень.

Також, є поділ на фіксовані та змінні. Фіксовані витрати залишаються незмінними незалежно від обсягів виробництва, наприклад, оренда, амортизація, адміністративні витрати. Змінні витрати змінюються разом з обсягом виробництва, наприклад, вартість сировини, електроенергії або заробітна плата робітників.

Постійні витрати – це витрати, які залишаються стабільними незалежно від змін у рівні діяльності компанії. Змінні витрати залежать від обсягів виробництва і можуть змінюватися пропорційно до цих обсягів.

Альтернативні витрати, або витрати втрачених можливостей, представляють вартість найкращого альтернативного варіанту використання ресурсів, який був упущений через вибір конкретного напрямку використання ресурсів.

Витрати можуть бути залежними від часу, наприклад, одноразовими або повторюваними, а також періодичними.

Вміння ефективно управляти витратами та отримувати від них максимальний результат є вирішальним чинником стабільності функціонування підприємства у ринкових умовах господарювання. [1]

Аналіз витрат є важливим інструментом для визначення прибутковості. Прибуток визначається як різниця між загальними доходами і загальними витратами. Мінімізація витрат при максимізації доходів є ключовою метою для будь-якого бізнесу. Витрати також впливають на процес ціноутворення. Компанії визначають ціни на свою продукцію на основі витрат, додавши до них маржу прибутку. Відповідно, знання структури витрат є важливим для встановлення конкурентоспроможних цін.

У контексті домогосподарств витрати можуть включати різноманітні категорії, такі як витрати на житло, харчування, транспорт, освіту, медичне обслуговування та інші поточні витрати. Управління витратами в домогосподарствах є важливим для досягнення фінансової стабільності та ефективного розподілу обмежених ресурсів.

«Доходи та витрати населення завжди займають важливе місце у житті населення та країни в цілому. Підвищення доходів та зменшення витрат домогосподарств є одною із найважливіших економічних проблем, на які звертає увагу уряд держави. Оскільки безпосередньо від даних показників залежить соціально-економічний стан країни та розвиток суспільства. Зі збільшенням доходів підвищуються якість життя та зростає споживання життєвих благ. Рівень життя характеризує темпи розвитку економіки, в Україні дана проблема особливо гостра.» [2]

Таким чином, витрати, як економічна категорія, є важливим інструментом для аналізу ефективності використання ресурсів, прийняття управлінських рішень і планування фінансової діяльності як у бізнесі, так і в особистих фінансах. Вони відображають необхідні витрати на виробництво та споживання, допомагають визначити прибутковість і впливають на процеси ціноутворення та прийняття рішень.

1.3 Соціально-економічна сутність процесу управління та аналізу витрат, ефективність їх застосування для домогосподарств

Сутність управління та аналізу витрат полягає в умінні ефективно розпоряджатися ресурсами для досягнення максимального результату за найменші витрати. Управління витратами – це процес планування, контролю та оптимізації витрат у підприємстві або організації. Цей процес включає в себе аналіз інформації про витрати, ідентифікацію ефективних стратегій зниження витрат, а також визначення способів оптимізації використання ресурсів.

На соціальному рівні управління витратами може впливати на рівень зайнятості, якість життя населення та розподіл ресурсів у суспільстві. Наприклад, ефективне управління витратами у галузі охорони здоров'я може покращити доступ до медичних послуг для населення та зменшити соціальну нерівність у цьому напрямку.

З економічної точки зору, управління витратами дозволяє підприємствам та організаціям знижувати виробничі витрати, підвищувати ефективність виробництва та конкурентоспроможність на ринку. Це може включати такі заходи, як оптимізація процесів виробництва, використання технологій для автоматизації, а також управління запасами та постачаннями.

Головна мета управління витратами полягає у забезпеченні того, щоб витрати не перевищували встановлений бюджет, а використання коштів було ефективним та раціональним. Це важливо як для домогосподарств, так і для організацій, оскільки контроль витрат допоможе уникнути фінансових труднощів та сприяти стабільному економічному зростанню.

Аналіз витрат відноситься до процесу збору, перегляду та оцінки даних про витрати для визначення шляхів оптимізації витрат. Аналіз включає дослідження того, де, як та чому гроші були витрачені, та визначення можливостей для зниження витрат без шкоди для продуктивності чи якості. Аналіз витрат допомагає приймати обґрунтовані фінансові рішення та може використовуватись для підтримки стратегічного планування.

Загалом, обидва ці процеси тісно переплетені і взаємозалежні, оскільки ефективно управління витратами не можливе без глибокого та систематичного аналізу витрат. Ці методики стають основою для створення стійкої економічної стратегії, яка може підтримувати як короткострокові, так і довгострокові фінансові цілі.

Узагальнюючи, можна підсумувати, що соціально-економічна сутність управління та аналізу витрат полягає в забезпеченні ефективного використання ресурсів для досягнення цілей підприємства, організації або суспільства в цілому, що сприяє зростанню ефективності, конкурентоспроможності та стійкому розвитку.

Так, і для домогосподарств, управління та аналіз витрат, є важливими аспектами їх фінансової стабільності та ефективного господарювання. Добре організоване домогосподарство здатне ефективно розподіляти ресурси, забезпечувати належний рівень життя та готуватися до можливих фінансових викликів.

Перший крок управління витратами – це створення бюджету, що відображає доходи та розподіл їх на різні категорії витрат. Бюджет може включати основні рахунки, такі як житло, їжа, транспорт, а також розділи для розваг, відпочинку, непередбачених витрат тощо.

Регулярний аналіз витрат дозволяє зрозуміти, куди саме йдуть гроші та як їх можна оптимізувати. Це може включати ідентифікацію зайвих витрат, пошук можливостей для зменшення витрат або перегляд пріоритетів витрат. Централізований аналіз витрат може виявити зони для зниження витрат і підвищення ефективності, що призводить до збільшення загальної економії.

Встановлення конкретних фінансових цілей для домогосподарства, такі як заощадження на автомобіль, оплата кредитів, покращення фінансової стабільності тощо, можуть служити мотивацією для зменшення витрат та ефективного управління бюджетом.

Управління витратами дозволяє прогнозувати майбутні фінансові ризики та забезпечувати стабільність бюджету, запобігаючи потенційні кризові ситуації.

Чітке управління витратами може забезпечити додаткові ресурси для інвестицій у довгострокові проекти, що покращують якість життя і сприяють економічному зростанню.

Якщо в домогосподарстві більше одного основного постачальника доходів, важливо залучити всіх членів родини до процесу управління витратами. Спільне прийняття рішень допомагає забезпечити згоду щодо фінансових цілей та стратегій їх досягнення.

Таким чином, управління витратами, постійний їх аналіз, є критично важливим компонентом для ефективного управління ресурсами домогосподарств.

1.4 Методи та принципи управління та аналізу витрат в домогосподарствах

Управління та аналіз витрат в домогосподарствах ґрунтується на методах та принципах, які допомагають раціонально використовувати сімейний бюджет та забезпечувати економічну ефективність.

Створення бюджету допомагає домогосподарству планувати та контролювати свої витрати.

Бюджет домогосподарства – це план розподілу доходів між різними категоріями витрат на побутові потреби та цілі сім'ї протягом певного періоду часу, як правило, місяця або року. Основні складові бюджету домогосподарства включають: доходи, витрати, заощадження, заборгованості.

Бюджет домогосподарства допомагає йому контролювати свої фінанси, уникати зайвих витрат, планувати на майбутнє та досягати своїх фінансових цілей. Він може бути складений вручну, використовуючи папір та олівець, або за допомогою різноманітних програм і мобільних додатків для фінансового планування.

Домогосподарства можуть виділяти кошти на основні категорії витрат, такі як харчування, житло, транспорт, розваги, заощадження тощо, і контролювати витрати в межах цих категорій (бюджетування).

Аналіз витрат допомагає домогосподарству зрозуміти, на що саме витрачаються гроші, він включає розбір витрат на компоненти для визначення необхідності та ефективності кожної статті витрат. Це може включати як виявлення надмірних витрат, так і ідентифікацію можливостей для зменшення витрат, перерозподіл ресурсів (аналіз витрат).

На основі результатів аналізу розробляються рекомендації щодо оптимізації витрат, удосконалення бізнес-процесів та підвищення ефективності використання ресурсів.

Оцінка ефективності використовується для аналізу того, наскільки ефективно грошові ресурси використовуються для досягнення фінансових цілей. Методи оцінки можуть включати розрахунок віддачі від інвестицій, аналіз рентабельності витрат та інші фінансові показники.

Домогосподарства можуть шукати способи для економії грошей, такі як покупка виключно необхідних товарів, використання купонів та знижок, раціональне використання енергії та води, а також уникнення зайвих витрат (ефективне використання ресурсів).

Планування витрат допомагає домогосподарствам розробити стратегію для досягнення довгострокових фінансових цілей, таких як накопичення на пенсію, забезпечення освіти для дітей, заощадження на випадок непередбачених обставин, визначати пріоритети щодо витрат. Це означає вибір того, що є найбільш важливим для сім'ї, і виділення коштів на ці потреби у першу чергу (планування та пріоритети).

Домогосподарства можуть розробляти стратегії для досягнення своїх фінансових цілей, таких як покупка житла, погашення боргів, накопичення заощаджень на майбутнє тощо (стратегічне планування).

Ці методи та принципи допомагають домогосподарствам керувати своїми витратами, забезпечуючи ефективне використання грошей і досягнення фінансової стійкості.

Управління фінансами домогосподарств включає планування та контроль витрат, бюджетування, заощадження та інвестування. Це важливо для досягнення

фінансової стабільності та забезпечення майбутніх потреб. Добре організоване домогосподарство здатне ефективно розподіляти ресурси, забезпечувати належний рівень життя та готуватися до можливих фінансових викликів.

2 ДОСЛІДЖЕННЯ ЗАСОБІВ ДЛЯ УПРАВЛІННЯ ТА АНАЛІЗУ ВИТРАТ У ДОМОГОСПОДАРСТВАХ

2.1 Аналіз інструментальних засобів для управління та аналізу витрат в домогосподарствах

На думку Т. Кізими «особисті фінанси вимагають управління завжди, а в нестабільних умовах – особливо», з чим дійсно можна погодитись, тому для ефективного їх використання необхідно знайти найбільш дієві інструменти для управління ними.

Для організації управління та аналізу витрат у домогосподарствах можуть бути використані такі інструментальні засоби:

- аналіз інформації подумки;
- паперові засоби;
- таблиці Excel, Google Sheets;
- мобільні та web-додатки.

Аналіз інформації подумки – найпоширеніший підхід і найменш ефективний. Намагання утримувати в пам'яті всі здійснені витрати, може бути складним через їх велику кількість; приблизне визначення залишків бюджету на основі пам'яті про витрачені та отримані кошти, спроба помітити та запам'ятати зміни у витратах з часом, все це може бути неточним через обмеженість людської пам'яті.

Паперовий облік витрат – це традиційний метод ведення обліку фінансів, який може бути корисним для тих, хто віддає перевагу ручному способу записування своїх витрат.

Це може бути простий щоденник або блокнот, в якому записуються витрати щодня. У ньому можливо створити розділи для різних категорій витрат, наприклад, їжа, транспорт, розваги тощо, і записувати кожну витрату відповідно до цих категорій.

Або це може бути створення енвелопів (конвертів) для готівки. Цей метод полягає в тому, що гроші розподіляються на різні енвелопи заздалегідь визначених категорій витрат. Наприклад, можливо створити окремі енвелопи для їжі, транспорту, розваг тощо. Кожного разу, коли здійснюються витрати, гроші беруться з відповідного енvelopа.

Ще можливо застосувати метод таблиць та графіків. При ньому використовуються аркуші паперу для ведення обліку витрат. Можливо створити таблиці для запису кожної витрати, а також графіки для візуалізації своїх витрат за певний період часу.

Також можливо зберігати всі квитанції та рахунки за покупками, а потім регулярно перевіряти їх і записувати витрати у відповідні журнали або таблиці.

Використання бюджетних планів передбачає створення бюджетного плану на папері, де зазначаються доходи та витрати на певний період часу. Потім відслідковується виконання цього плану – записування всіх витрат та порівняння їх з запланованими сумами.

Ці методи можуть бути корисними для тих, хто віддає перевагу традиційним методам ведення обліку фінансів і бажає зберігати контроль над своїми витратами за допомогою ручних записів. Однак важливо мати на увазі, що ці методи є трудомісткими та потребують великої сконцентрованості та уважності.

Таблиці Excel та Google Sheets можуть також використовуватись як інструменти для організації управління та аналізу витрат у домогосподарствах. Вони надають зручний спосіб вести облік фінансів, створювати звіти та аналізувати різні аспекти бюджету. Ось деякі способи, як можна використовувати Excel або Google Sheets для керування витратами:

- створення бюджету: можливо створити таблицю, де зазначаються різні категорії витрат (їжа, транспорт, розваги тощо) і встановлюються щомісячні бюджетні ліміти для кожної категорії. Потім записуються витрати щодня або щотижня і відслідковуються, як вони відповідають бюджету;

- аналіз витрат: використовуються функції Excel або Google Sheets, створюються різноманітні звіти і графіки, що візуалізують витрати. Наприклад,

можна побудувати графік кругової діаграми (рис. 2.1), щоб побачити, як різні категорії витрат відображаються на загальній сумі;

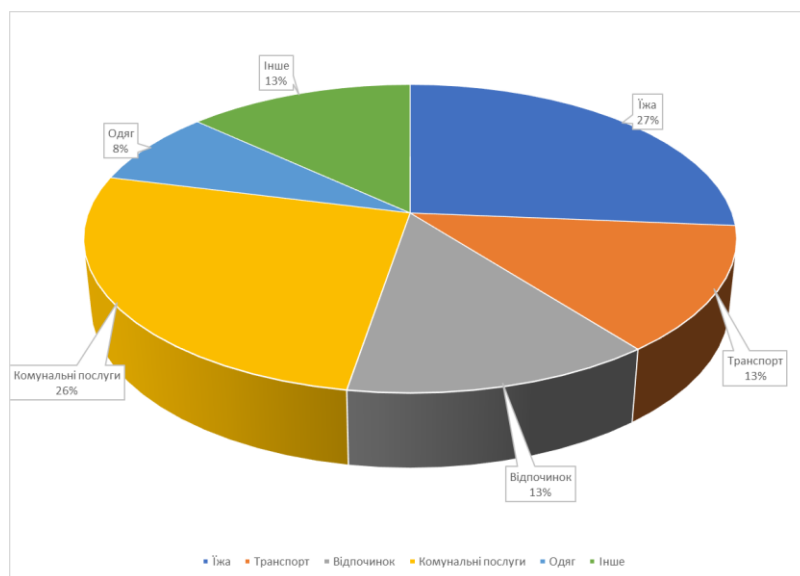


Рис. 2.1 Кругова діаграма витрат Excel

- автоматизація обробки даних: використовуються функції Excel або Google Sheets, такі як функції SUMIF або AVERAGE, для автоматичного обчислення сум витрат за кожною категорією або середніх значень витрат за певний період часу;

- спільна робота: Google Sheets надає можливість спільно працювати над таблицями з іншими членами домогосподарства. Можна ділитися таблицями зі своїми родичами і спільно вести облік витрат та планувати бюджет.

Таблиці Excel і Google Sheets мають свої недоліки. Це і складність використання для початківців; і час, необхідний на підтримку; і ризик помилок; обмеження щодо доступності та кількість користувачів; і потенційна втрата даних.

Мобільні та веб-додатки є дієвими інструментами для організації управління та аналізу витрат у домогосподарствах. Вони надають зручний та ефективний спосіб вести облік фінансів, відстежувати витрати та аналізувати їх у реальному часі.

Розглянемо деякі з них детальніше:

Personal Capital – це популярний додаток для управління особистими фінансами, який поєднує інструменти для відстеження витрат і бюджетування з потужними функціями для управління інвестиціями. Додаток надає користувачам

можливість отримати повний огляд їхнього фінансового стану, включаючи банківські рахунки, кредитні картки, іпотеку, позики та інвестиційні портфелі.

Однією з головних особливостей Personal Capital є зручний інтерфейс, який забезпечує інтеграцію з багатьма фінансовими установами, що дозволяє автоматично імпортувати транзакції і постійно оновлювати фінансові дані. Це дозволяє користувачам бачити всю свою фінансову картину в одному місці, що спрощує управління грошима і прийняття фінансових рішень.

Personal Capital також пропонує потужні інструменти для аналізу інвестицій. Користувачі можуть відстежувати продуктивність своїх портфелів, аналізувати розподіл активів і оцінювати ризики. Додаток надає докладну інформацію про витрати на інвестиції, включаючи комісії та збори, що допомагає користувачам знижувати витрати і підвищувати прибутковість своїх інвестицій.

Додаток також включає функції для планування пенсійних заощаджень. Personal Capital допомагає користувачам визначати, чи достатньо вони заощаджують для виходу на пенсію, і пропонує інструменти для моделювання різних сценаріїв пенсійних планів. Це дозволяє користувачам планувати свої заощадження з урахуванням їхніх фінансових цілей і потреб.

Крім того, Personal Capital пропонує послуги фінансового консультування, надаючи доступ до сертифікованих фінансових консультантів, які можуть допомогти користувачам розробити персоналізовані фінансові плани та стратегії інвестування. Це робить додаток особливо корисним для тих, хто шукає професійну підтримку у управлінні своїми фінансами.

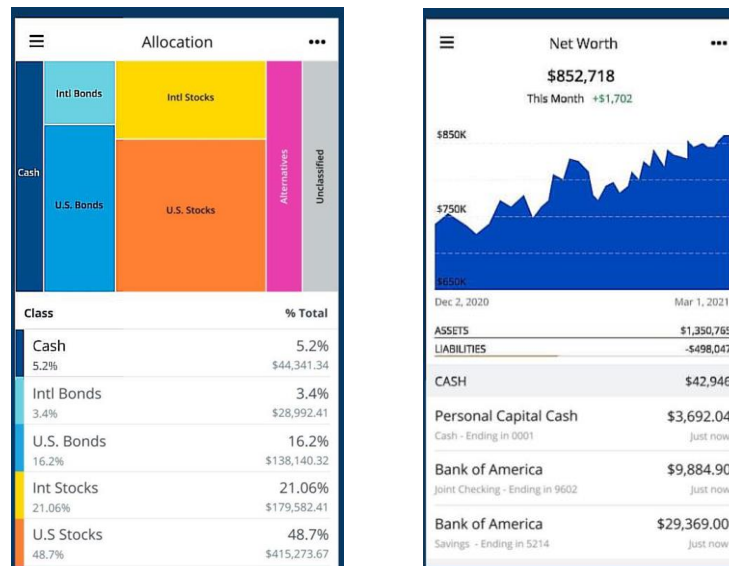


Рис. 2.2 Приклади екранних форм додатку Personal Capital

PocketGuard – це мобільний додаток для управління особистими фінансами, який допомагає користувачам контролювати свої витрати та зберігати більше грошей. Він відзначається простим та інтуїтивно зрозумілим інтерфейсом, що робить його доступним для широкого кола користувачів, незалежно від рівня фінансової грамотності.

Одна з головних особливостей PocketGuard – це автоматичне імпортування транзакцій з банківських рахунків і кредитних карток. Додаток інтегрується з багатьма фінансовими установами, що дозволяє постійно оновлювати фінансові дані користувача. Це забезпечує повний огляд фінансового стану в реальному часі, що допомагає користувачам приймати обґрунтовані фінансові рішення.

PocketGuard має функцію "In My Pocket", яка показує, скільки грошей залишається у користувача після врахування основних витрат, рахунків і заощаджень. Ця функція допомагає користувачам контролювати свої витрати та уникати перевитрат, що є важливим для досягнення фінансових цілей і підтримання бюджету.

Безпека даних є пріоритетом для PocketGuard. Додаток використовує шифрування та інші сучасні засоби захисту для забезпечення конфіденційності фінансової інформації користувачів. Це дозволяє користувачам відчувати себе впевнено при використанні додатку для управління своїми фінансами.



Рис. 2 Приклади екранних форм додатку PocketGuard

Додаток YNAB (You Need A Budget) спонукає користувачів навчитись ефективному управлінню своїми фінансами. Він доступний у вигляді безкоштовної пробної версії, яка триває 34 дні, та платної версії. YNAB має основний веб-додаток і мобільні додатки для різноманітних платформ, включаючи iPhone, Android, iPad, Apple Watch і Alexa, а також API для розробки додаткових функцій.

Додаток базується на чотирьох основних правилах бюджетування, які допомагають користувачам змінити свої фінансові звички і краще управляти своїми грошима.

Перше правило YNAB – "Надайте кожному долару роботу". Це означає, що користувачі повинні розподілити всі свої доходи між різними категоріями витрат, заощаджень та боргів, таким чином визначаючи, як кожен долар буде витрачений або збережений. Це допомагає уникнути непередбачених витрат і краще контролювати фінансові потоки.

Друге правило – "Збережіть на дощовий день". YNAB заохочує користувачів створювати резервні фонди для непередбачених витрат, таких як медичні витрати або ремонт автомобіля. Це допомагає підготуватися до фінансових несподіванок і зменшити стрес, пов'язаний з раптовими витратами.

Третє правило – "Перенесіть свої витрати". Це правило допомагає користувачам зрозуміти, що вони можуть переносити витрати між різними категоріями бюджету, якщо витрати перевищують заплановані ліміти. Це дозволяє гнучко реагувати на зміни у фінансовій ситуації та залишатися в рамках загального бюджету.

Четверте правило – "Витрачайте минуломісячні доходи". YNAB рекомендує користувачам жити на гроші, зароблені в попередньому місяці, що допомагає створити фінансову подушку і уникнути життя від зарплати до зарплати. Це правило сприяє більшому фінансовому стабільності і дозволяє краще планувати витрати.

У YNAB бюджетування ґрунтується на принципі встановлення цілей, що зазвичай обмежують витрати, і користувачі стежать за своїми випадковими витратами та перевіряють поточні доходи і витрати, щоб досягти цих цілей. Протягом місяця користувачам потрібно вносити свої витрати і в разі потреби коригувати бюджет, перерозподіляючи кошти між категоріями у випадку непередбачуваних витрат. Усі фінансові дані можна візуалізувати у формі графіків і діаграм, які користувачі можуть налаштовувати.

Захист даних у YNAB забезпечується за допомогою шифрування і хешування паролів, а також використанням сертифікованих центрів обробки даних. Одним із недоліків є те, що мобільний додаток можна використовувати лише з встановленою десктопною версією. Додаткові функції, такі як sms-сповіщення про перевищення бюджету чи створення звітів за індивідуальним шаблоном, доступні за додаткову плату і можуть вимагати додаткових програмних рішень.

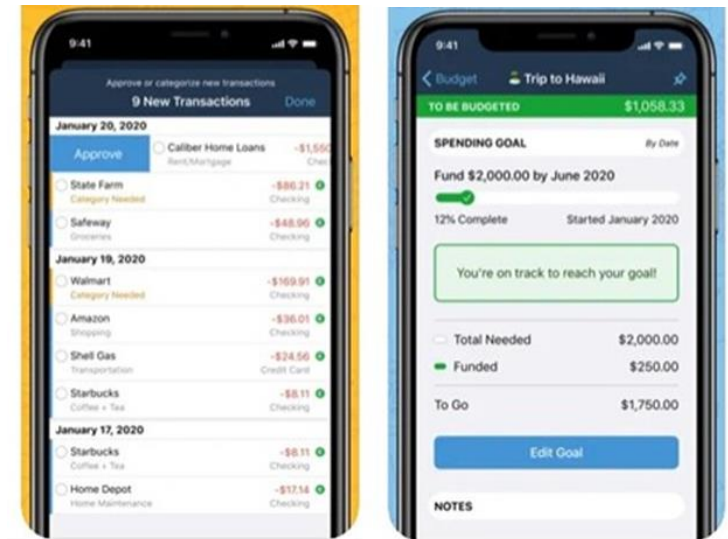


Рис. 2.4 Приклади екранних форм додатку YNAB

Зведені результати аналізу характеристик розглянутих додатків наведено у таблиці 2.1.

Таблиця 2.1

Аналіз характеристик аналогів для управління та аналізу витрат

Показник	PocketGuard	Personal Capital	YNAB	ВАШЕ ФІНАНСОВЕ ЗДОРОВ'Я (розроблений додаток)
Категорії витрат	фіксований список категорій	фіксований список категорій	фіксований список категорій	є можливість додавання власних категорій
Планування бюджету	+	+	+	+

Продовження таблиці 2.1

Аналіз характеристик аналогів для управління та аналізу витрат

Показник	PocketGuard	Personal Capital	YNAB	ВАШЕ ФІНАНСОВЕ ЗДОРОВ'Я (розроблений додаток)
Розподіл бюджету за категоріями	-	-	-	+
Визначення термінів бюджету	термін фіксований (30, 60, 90 днів)	термін фіксований (30, 60, 90 днів)	термін фіксований (30, 60, 90 днів)	є можливість самостійно обрати граничні дати
Графічні звіти витрат	+	-	+	+
Експорт фінансових даних	-	-	-	+
Українська локалізація	-	-	-	+

Таблиця демонструє, що деякі функції присутні лише у розробленому додатку (розподіл бюджету за категоріями, експорт фінансових даних, українська локалізація), а деякі функції покращено у розробленому додатку (управління категоріями витрат, визначення термінів бюджету, графічні звіти витрат).

2.2 Автоматизоване управління та аналіз витрат в домогосподарствах як засіб для підвищення ефективності їх господарювання, його переваги

Автоматизоване управління та аналіз витрат в домогосподарствах – це важливий аспект сучасного фінансового планування, який дозволяє

домогосподарствам ефективно керувати своїми фінансами за допомогою технологічних засобів.

Автоматизоване управління витратами включає в себе використання програмного забезпечення та інших технологічних інструментів для моніторингу, аналізу, планування та контролю фінансових ресурсів домогосподарства. Це дозволяє домогосподарствам відстежувати свої доходи та витрати, встановлювати бюджети, прогнозувати майбутні фінансові потреби та ідентифікувати можливості для зниження непотрібних витрат.

Воно може включати в себе автоматизовані системи для відстеження доходів та витрат, програми для бюджетування, а також інтелектуальні системи для аналізу споживчої поведінки.

Переваги автоматизованого управління та аналізу витрат:

економія часу та зусиль:

- програмне забезпечення може автоматизувати багато рутинних завдань, пов'язаних з управлінням фінансами;

краще розуміння структури витрат:

- аналіз даних може допомогти зрозуміти, куди йдуть гроші, та де можлива економія;

ефективне планування та контроль бюджету:

- автоматизоване управління бюджетом може допомогти дотримуватися своїх фінансових цілей;

досягнення фінансових цілей:

- завдяки кращому розумінню своїх витрат та ефективному плануванню з'являється можливість швидше досягати своїх фінансових цілей.

Тобто, *сутність автоматизованого управління та аналізу витрат* полягає в тому, що: уся інформація про фінанси знаходиться в одному місці, це робить її легкодоступною для аналізу (централізація даних); програмне забезпечення може автоматично генерувати звіти, що допомагає зрозуміти витрати (автоматизація аналізу); є можливість налаштування системи відповідно до своїх потреб та цілей (персоналізація).

Автоматизоване управління та аналіз витрат в домогосподарствах – це процес використання програмного забезпечення для:

- збору даних про всі фінансові транзакції, що здійснюються членами домогосподарства;
- категоризації цих транзакцій за певними категоріями (наприклад, продукти харчування, транспорт, розваги);
- аналіз даних для виявлення тенденцій, закономірностей та проблемних зон;
- планування майбутніх витрат та досягнення фінансових цілей.

Задача автоматизованого управління та аналізу витрат полягає в розробці системи, що дозволить користувачам ефективно управляти своїми витратами, автоматично аналізувати фінансову інформацію та вносити зміни в бюджет відповідно до їх потреб і пріоритетів. Все це, має на меті мінімізувати ручну роботу, знизити витрати та оптимізувати бюджетне планування.

Задача автоматизації управління та аналізу витрат в домогосподарствах може бути вирішена за допомогою різних інформаційно-комунікаційних засобів і технологій. Ось деякі можливі шляхи вирішення цієї задачі:

Розробка спеціалізованого програмного забезпечення: створення програм для керування фінансами, які враховують потреби домогосподарств. Ці програми повинні бути простими у використанні та мати зручний інтерфейс для введення та аналізу фінансових даних.

Використання веб-додатків: це зручно для домогосподарств, які хочуть мати доступ до своїх фінансових даних у будь-який час і з будь-якого місця.

Використання мобільних додатків: розробка мобільних додатків для відстеження витрат і керування фінансами. Це особливо корисно для активних людей, які багато часу проводять поза домом і хочуть мати можливість відстежувати свої витрати, що називається «на ходу».

Використання інтернет-банкінгу: використання сервісів інтернет-банкінгу для автоматичного відстеження та аналізу фінансових транзакцій. Більшість банків пропонують інтерфейси для моніторингу фінансових операцій та створення звітів.

Цільовою аудиторією продукту є домогосподарства будь-якого розміру, які бажають краще керувати своїми фінансами. Вони можуть висувати такі вимоги до вирішення цієї задачі:

- простота використання: для багатьох користувачів важливо, щоб програма або сервіс був легко зрозумілим і простим у використанні;
- доступність: домогосподарства хочуть мати доступ до своїх фінансових даних у будь-який час і з будь-якого місця, тому важливо, щоб програма або сервіс був доступний онлайн і на мобільних пристроях;
- функціональність: домогосподарства можуть мати різні потреби стосовно керування фінансами, тому важливо, щоб програма або сервіс мав широкий спектр функцій.

На важливості інтеграції автоматизації в системи управління фінансами наголошують багато дослідників, такі як, Грабар О. І., Остроухов М. С., Постова С. А., Лось В.О., Максишко Н.К., Столярчук І.А., Лукановська І.Р., Лошенко О.В., підкреслюючи, що це може значно підвищити ефективність фінансового планування.

Для мінімізації допущених помилок, втрат та неефективного використання ресурсів домогосподарствам потрібна допомога. Такою допомогою може стати програмне забезпечення для автоматизованого управління та аналізу витрат в домогосподарствах.

На сьогодні існує ряд програмних продуктів, що допомагають користувачам вести облік особистих фінансів. Проте, багато з них не забезпечують повного аналізу витрат, є складними у використанні, що ніяк не мотивує користувачів до переходу на автоматизоване управління та аналіз особистих витрат.

3 ПОСТАНОВКА ТЕХНІЧНОГО ЗАВДАННЯ ТА ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ТА АНАЛІЗУ ВИТРАТ В ДОМОГОСПОДАРСТВАХ

3.1 Визначення вимог до програмного забезпечення

Аналіз функціональних та нефункціональних вимог дозволяє всебічно зрозуміти, які функції та характеристики повинна мати система, щоб задовольнити потреби користувачів та забезпечити ефективну роботу. Це важливий крок у процесі розробки додатку, який допомагає створити якісний продукт, що відповідає очікуванням користувачів і вимогам ринку.

Функціональні вимоги описують конкретні дії, які система повинна виконувати, вони визначають функціональні можливості додатку. Нефункціональні вимоги визначають характеристики системи, які впливають на її якість та ефективність.

Функціональні вимоги до додатку:

1. Можливість реєстрації та авторизації.
2. Можливість введення користувачем доходів та витрат.
3. Можливість додавання та видалення категорій.
4. Можливість створення бюджету.
5. Можливість створення бюджету із зазначенням початкової та кінцевої дати.
6. Можливість генерування графічного звіту витрат.

Нефункціональні вимоги.

1. Локалізація українською мовою.
2. Сумісність з сучасними веб-браузерами (Chrome, Firefox, Safari, Edge, Opera).
3. Підтримка різних розмірів екрану (ПК, планшети, смартфони).

3.2 Засоби реалізації програмного забезпечення для автоматизованого управління та аналізу витрат в домогосподарствах

Python – це широко використовувана мова програмування, яка відзначається простотою та читабельністю коду. Вона має інтуїтивно зрозумілий синтаксис, який нагадує природну мову, що робить Python легким для вивчення і використання, навіть для початківців. Він абстрагується від багатьох деталей апаратного забезпечення, що дозволяє програмістам зосередитися на вирішенні логічних задач, а не на низькорівневих аспектах програмування.

Python є інтерпретованою мовою, що означає виконання коду рядок за рядком. Це спрощує тестування та налагодження коду, хоча може впливати на продуктивність порівняно з компільованими мовами. Однією з ключових переваг Python є його кросплатформенність, оскільки програми можуть виконуватися на різних операційних системах без змін у коді (Windows, macOS, Linux тощо), що забезпечує велику гнучкість та зручність для розробників.

Мова має величезну стандартну бібліотеку, яка включає безліч модулів і пакетів для виконання різноманітних завдань (робота з файлами, мережева взаємодія, робота з даними, веб-розробка тощо). Це дозволяє розробникам швидко знаходити та використовувати необхідні інструменти. Python також підтримує об'єктно-орієнтоване програмування, що дозволяє створювати складні програми, використовуючи об'єкти і класи, що сприяє повторному використанню коду та полегшує його підтримку.

Однією з найбільших переваг Python є широка спільнота та підтримка. Мова має одну з найбільших і найактивніших спільнот розробників, що надає безліч ресурсів для навчання, включаючи документацію, підручники, форуми та обговорення на платформах, таких як Stack Overflow. Гнучкість та розширюваність Python дозволяють інтегрувати її з іншими мовами програмування, такими як C, C++ та Java, що розширює функціональність існуючих систем. Підтримка різних парадигм програмування (процедурне, функціональне та об'єктно-орієнтоване

програмування) робить Python універсальним інструментом для вирішення широкого спектра завдань.

Python використовується у веб-розробці (Django, Flask), науці про дані (Pandas, NumPy, SciPy), машинному навчанні (TensorFlow, Keras), автоматизації (Selenium), ігровій розробці (Pygame) та багатьох інших галузях.

Однак Python має і свої недоліки. Як інтерпретована мова, він може бути повільнішим у виконанні порівняно з компільованими мовами, такими як C або Java. Крім того, Python може використовувати більше пам'яті через свою високу абстракцію та автоматичне керування пам'яттю.

Але все ж таки, Python – це ідеальний вибір для розробників різного рівня і для широкого спектра задач.

Flask – це легкий веб-фреймворк для Python, який використовується для створення веб-додатків і API. Його основні характеристики роблять його популярним вибором серед розробників, особливо для невеликих проектів та прототипів, він є популярним вибором для стартапів і невеликих команд розробників, які хочуть швидко вивести свої ідеї на ринок. Він також часто використовується для навчальних цілей, оскільки дозволяє швидко освоїти основи веб-розробки.

Flask є мікрофреймворком, що означає, що він надає мінімальний набір базових функцій, залишаючи розробникам можливість вибирати додаткові компоненти та бібліотеки залежно від потреб проекту. Ця гнучкість дозволяє створювати прості додатки без зайвих ускладнень, але також дає змогу масштабувати їх, додаючи необхідні функціональні можливості.

Фреймворк побудований на основі інструментів Werkzeug і шаблонного двигуна Jinja2. Werkzeug забезпечує роботу з HTTP-запитами та управління маршрутизацією, тоді як Jinja2 дозволяє створювати динамічні HTML-сторінки з використанням шаблонів. Flask є дуже легким у налаштуванні та використанні, що робить його ідеальним для швидкої розробки і тестування. Він забезпечує простий інтерфейс для налаштування маршрутів (URL-адрес) і обробників (функцій, які

відповідають на запити до цих маршрутів), що дозволяє швидко створювати веб-інтерфейси і API.

Flask надає лише базові компоненти, необхідні для створення веб-додатків, що дозволяє розробникам додавати лише ті функції, які потрібні для конкретного проекту. Гнучкість Flask дозволяє вибирати і інтегрувати різні розширення та бібліотеки для додавання нових можливостей, таких як аутентифікація, ORM (Object-Relational Mapping), валідація форм тощо. Простота використання фреймворка, зрозуміла документація та простий у використанні API роблять його доступним для початківців. Використання шаблонного двигуна Jinja2 дозволяє створювати динамічні HTML-сторінки, які легко інтегруються з Python-кодом. Flask підтримує модульність додатків, дозволяючи організовувати код у окремі модулі або пакети, що полегшує масштабування і підтримку великих проектів.

Втім, у порівнянні з більш повнофункціональними фреймворками, такими як Django, Flask не має вбудованих компонентів для обробки багатьох типових задач, що може вимагати більше ручної роботи і вибору сторонніх бібліотек. Хоча Flask може бути масштабованим, його легка архітектура може стати викликом при роботі над дуже великими проектами, де більш структуровані фреймворки можуть бути кращими. Flask не має вбудованих систем для управління базами даних, аутентифікації користувачів, обробки форм тощо, що потребує додаткових зусиль для інтеграції необхідних компонентів.

Pandas – це потужна бібліотека для обробки та аналізу даних у мові програмування Python. Вона забезпечує високоефективні, легкі у використанні структури даних та інструменти для роботи з табличними даними та часовими рядами. Pandas надає два основних типи даних: Series і DataFrame. Series – це одновимірний масив із мітками (індексом), подібний до списку або одновимірного масиву NumPy. DataFrame – це двовимірна таблична структура, подібна до таблиці в базі даних або електронній таблиці, що складається з рядів та стовпців.

Одна з ключових переваг Pandas – це здатність обробляти великі обсяги даних ефективно і швидко. Вона дозволяє виконувати складні маніпуляції з даними, такі як злиття, сортування, фільтрація та групування, з мінімальним

обсягом коду. Інтеграція з іншими бібліотеками Python, такими як NumPy та Matplotlib, робить Pandas ще потужнішим інструментом для наукових обчислень та візуалізації даних.

Переваги використання Pandas включають простоту використання завдяки інтуїтивно зрозумілому API, що робить роботу з даними зручною та ефективною. Гнучкість Pandas дозволяє легко маніпулювати даними, виконуючи операції над рядками та стовпцями, злиття, об'єднання, перетворення даних та багато іншого. Бібліотека побудована на основі NumPy, що забезпечує високу продуктивність при роботі з великими наборами даних. Pandas легко інтегрується з іншими науковими бібліотеками Python, такими як NumPy, SciPy та Matplotlib, що дозволяє створювати потужні аналітичні додатки. Вона також надає широкий спектр функцій для роботи з часовими рядами, включаючи обробку дат і часу, ресемплінг та агрегацію.

Проте, Pandas має і свої недоліки. Вона може споживати велику кількість пам'яті при роботі з дуже великими наборами даних, що може бути проблемою для обмежених ресурсів. Хоча Pandas дуже ефективна для багатьох завдань, вона може бути менш продуктивною для деяких спеціалізованих обчислень порівняно з більш низькорівневими бібліотеками. Крім того, Pandas має багатий набір функцій, які можуть бути складними для освоєння новачками.

Pandas широко використовується в наукових дослідженнях, фінансовому аналізі, машинному навчанні, бізнес-аналітиці та інших сферах, де потрібна обробка та аналіз великих обсягів даних. Вона дозволяє швидко і ефективно виконувати такі завдання, як завантаження даних з різних форматів (CSV, Excel, SQL, JSON), очищення та перетворення даних, обчислення статистичних показників, створення візуалізацій та багато іншого.

SQLAlchemy – це популярна бібліотека для Python, яка надає потужний і гнучкий інструмент для роботи з базами даних. Вона підтримує як SQL, так і ORM (Object-Relational Mapping) підходи, що дозволяє розробникам вибирати найкращий спосіб взаємодії з базами даних відповідно до потреб проекту. SQLAlchemy надає два основні компоненти: SQLAlchemy Core і SQLAlchemy

ORM. SQLAlchemy Core – це низькорівнева бібліотека для роботи з SQL-запитами, яка дозволяє створювати, виконувати та керувати запитами до бази даних, використовуючи вирази SQL. SQLAlchemy ORM – це високорівнева бібліотека, що забезпечує мапування об'єктів Python на записи в базі даних, дозволяючи працювати з даними як з об'єктами.

Однією з головних переваг SQLAlchemy є її гнучкість і потужність. Вона підтримує різні бази даних, включаючи PostgreSQL, MySQL, SQLite, Oracle та багато інших. SQLAlchemy також дозволяє використовувати різні стилі взаємодії з базами даних, від прямого написання SQL-запитів до використання ORM для роботи з об'єктами Python. SQLAlchemy забезпечує потужні інструменти для управління транзакціями, з'єднаннями та сесіями, що дозволяє ефективно керувати взаємодією з базою даних. Вона також підтримує складні операції, такі як об'єднання таблиць, агрегації, фільтрація та багато інших, що дозволяє виконувати складні запити з мінімальними зусиллями.

Переваги SQLAlchemy включають простоту використання завдяки зрозумілому API, що робить роботу з базами даних ефективною та інтуїтивною. Гнучкість SQLAlchemy дозволяє розробникам вибирати між використанням SQL або ORM, залежно від вимог проекту. Підтримка різних баз даних забезпечує широкі можливості для інтеграції з існуючими системами. Потужні інструменти для управління транзакціями та сесіями забезпечують надійне та ефективне керування взаємодією з базою даних.

Проте, SQLAlchemy має і свої недоліки. Одним з них є складність освоєння для новачків, оскільки вона пропонує багатий набір функцій і можливостей, які можуть бути складними для розуміння на початкових етапах. Крім того, використання ORM може призвести до зниження продуктивності при роботі з великими обсягами даних або складними запитам.

SQLAlchemy широко використовується у веб-розробці, наукових дослідженнях, аналізі даних та інших сферах, де необхідна взаємодія з базами даних. Вона дозволяє швидко і ефективно виконувати такі завдання, як створення, читання, оновлення та видалення даних, об'єднання таблиць, агрегації, фільтрація

та багато іншого. Завдяки своїй гнучкості та потужності, SQLAlchemy є незамінним інструментом для розробників, які працюють з базами даних у Python.

SQLite – це легка, вбудована, реляційна база даних, яка є однією з найпопулярніших баз даних у світі завдяки своїй простоті, надійності та ефективності. Вона є сервер-less базою даних, що означає, що не потребує окремого серверного процесу для роботи. Всі дані зберігаються в одному файлі на диску, що робить її ідеальною для вбудованих систем та локальних додатків, оскільки налаштування та використання спрощуються через відсутність потреби в окремому сервері.

Однією з головних переваг SQLite є її легкість. Вона має невеликий розмір і не потребує великої кількості ресурсів, що робить її ідеальною для використання в мобільних додатках, вбудованих системах та інших середовищах з обмеженими ресурсами. Незважаючи на свій малий розмір, SQLite підтримує більшість стандартних SQL-операцій, включаючи транзакції, об'єднання, агрегації, індексацію та інші складні запити. Вона є надійною базою даних, яка забезпечує повну підтримку ACID-транзакцій (Atomicity, Consistency, Isolation, Durability), що гарантує надійність і безпеку операцій навіть у разі збою системи або несподіваного вимкнення живлення.

Ключовою особливістю SQLite є її портативність. Всі дані зберігаються в одному файлі, який можна легко переміщувати між різними платформами та пристроями, що робить її зручною для резервного копіювання, відновлення та обміну даними між різними системами. Основні переваги SQLite включають простоту використання через відсутність необхідності в налаштуванні окремого серверного процесу, легкість і невеликий розмір, що дозволяє використовувати її в середовищах з обмеженими ресурсами, а також надійність і підтримку ACID-транзакцій, які забезпечують безпеку і надійність операцій з даними. Портативність даних дозволяє легко переміщувати базу даних між різними платформами і пристроями.

Однак, SQLite має і свої недоліки. Вона не підходить для високонавантажених серверних додатків, які потребують обробки великої кількості

одночасних запитів або мають великі обсяги даних. У таких випадках краще використовувати більш потужні серверні бази даних, такі як PostgreSQL або MySQL. Крім того, обмеженість у підтримці деяких розширених функцій SQL може бути проблемою для складних додатків.

SQLite широко використовується в мобільних додатках, вбудованих системах, веб-браузерах, іграх, настільних програмах та багатьох інших областях. Вона ідеально підходить для розробки додатків, які потребують легкої та надійної локальної бази даних.

JavaScript – це ключова мова програмування для розробки фронтенд-частини веб-додатків, що дозволяє створювати багатофункціональні, інтерактивні, динамічні та зручні веб-інтерфейси. Ця динамічна, інтерпретована мова програмування є основним інструментом для створення сучасних веб-сторінок. JavaScript дозволяє маніпулювати елементами DOM (Document Object Model), створюючи динамічні й зручні інтерфейси. За допомогою JavaScript можна реалізувати широкий спектр функцій, від простих інтерактивних елементів до складних односторінкових додатків (SPA).

JavaScript виконується на стороні клієнта, тобто безпосередньо у веб-браузері користувача, що дозволяє швидко реагувати на дії користувача без необхідності звертатися до серверу. Це забезпечує високу швидкість і плавність роботи інтерфейсу. Мова підтримує об'єктно-орієнтоване програмування, що дозволяє створювати модульний і повторно використовуваний код, а також функціональне програмування, що додає гнучкості в написанні коду.

JavaScript має велику кількість популярних бібліотек та фреймворків, таких як React, Angular, Vue.js, які значно спрощують і прискорюють розробку складних інтерфейсів. Ці фреймворки надають розробникам інструменти для створення компонентних систем, управління станом додатка та взаємодії з сервером. Мова також підтримує асинхронне програмування за допомогою таких механізмів, як колбеки, проміси та асинхронно-очікуючі функції (`async/await`). Це дозволяє обробляти асинхронні запити до серверу, наприклад, для завантаження даних без перезавантаження сторінки, що підвищує користувацький досвід.

JavaScript дозволяє створювати динамічні й інтерактивні елементи на веб-сторінках, забезпечує швидку реакцію на дії користувача, підтримує різні парадигми програмування, має широкий вибір інструментів для розробки і підтримує асинхронні запити до серверу. Однак, як клієнтська мова, JavaScript може бути вразливим до атак, таких як XSS (Cross-Site Scripting). Інтерпретована природа може впливати на швидкість виконання порівняно з компільованими мовами, а відмінності в реалізації JavaScript у різних браузерах можуть призводити до проблем сумісності.

HTML (HyperText Markup Language) – це основна мова розмітки для створення та структурування веб-сторінок. Вона визначає зміст і структуру документів, які відображаються в браузері, використовуючи теги та атрибути для опису різних елементів на сторінці.

HTML є основою веб-дизайну і розробки, забезпечуючи базову структуру для тексту, зображень, відео, посилань, форм та інших елементів. Вона працює разом з CSS (Cascading Style Sheets) і JavaScript, які забезпечують стилізацію та інтерактивність веб-сторінок відповідно.

Однією з ключових характеристик HTML є її простота та легкість у вивченні. HTML використовує систему тегів, які обмежують різні частини контенту. Теги зазвичай приходять у парі: відкриваючий тег (наприклад, `<p>`) і закриваючий тег (наприклад, `</p>`), де "p" означає абзац. HTML також підтримує самозакриваючі теги, такі як `` для зображень.

HTML є кросплатформенною мовою, яка підтримується всіма сучасними веб-браузерами. Це означає, що веб-сторінки, створені за допомогою HTML, можуть бути переглянуті на будь-якому пристрої з браузером, включаючи комп'ютери, планшети та смартфони.

HTML також забезпечує семантичну структуру документів, що поліпшує доступність та SEO (оптимізацію для пошукових систем). Семантичні теги, такі як `<header>`, `<footer>`, `<article>`, і `<section>`, надають додаткову інформацію про структуру та зміст веб-сторінок, що допомагає пошуковим системам і допоміжним технологіям краще розуміти і взаємодіяти з контентом.

Переваги використання HTML включають простоту використання, широку підтримку браузерами, кросплатформенність і можливість створення структурованого та семантичного контенту. Проте HTML має і свої обмеження. Наприклад, він не підтримує динамічні функціональні можливості, такі як обробка подій чи зміна контенту без перезавантаження сторінки – для цього використовується JavaScript. Також HTML не забезпечує стилізацію елементів, що є завданням CSS.

HTML широко використовується в веб-розробці для створення всіх типів веб-сторінок і додатків. Вона є основою для всіх інших технологій, які використовуються для веб-розробки, забезпечуючи базову структуру та розмітку контенту.

CSS (Cascading Style Sheets) – це мова стилів, що використовується для опису зовнішнього вигляду веб-сторінки, написаної мовою розмітки HTML або XML. CSS дозволяє відокремлювати контент веб-сторінки від її представлення, що робить розробку і підтримку веб-сайтів більш ефективною.

CSS використовується для визначення стилів елементів HTML, таких як кольори, шрифти, відступи, розміри, вирівнювання та розташування. Завдяки цьому можна створювати естетично привабливі та функціональні веб-сторінки. CSS підтримує різні методи для застосування стилів до HTML-елементів: внутрішні стилі (вбудовані в HTML-код), зовнішні стилі (вказані в окремих CSS-файлах) та вбудовані стилі (застосовані безпосередньо до HTML-елементів через атрибут style).

Однією з головних характеристик CSS є каскадність. Каскадність дозволяє застосовувати кілька стилів до одного елемента HTML, причому стиль з вищим пріоритетом замінює стиль з нижчим пріоритетом. Пріоритет визначається специфічністю селекторів та їх положенням у коді. Це дозволяє створювати гнучкі та складні стилі для різних частин веб-сторінки.

CSS також підтримує медіа-запити, що дозволяє створювати адаптивні дизайни для різних пристроїв і екранних розмірів. За допомогою медіа-запитів можна змінювати стилі веб-сторінки залежно від розміру екрана, орієнтації

пристрою та інших параметрів. Це дозволяє створювати веб-сторінки, які добре виглядають і функціонують як на комп'ютерах, так і на мобільних пристроях.

Переваги CSS включають можливість відокремлювати контент від стилів, що спрощує підтримку та оновлення веб-сторінок. Використання зовнішніх CSS-файлів дозволяє повторно використовувати стилі на кількох сторінках, що зменшує обсяг коду і підвищує продуктивність. CSS також підтримує складні макети та анімації, що дозволяє створювати динамічні та інтерактивні веб-сторінки.

Проте, CSS має і свої недоліки. Наприклад, різні браузерери можуть по-різному інтерпретувати CSS-стилі, що може призводити до проблем із сумісністю.

CSS широко використовується в веб-розробці для створення всіх типів веб-сторінок і додатків. Вона є ключовою технологією для створення візуально привабливих і функціональних веб-дизайнів.

PyCharm – це потужне інтегроване середовище розробки (IDE) для Python, розроблене компанією JetBrains. Воно пропонує безліч функцій, що роблять процес розробки зручнішим, ефективнішим і продуктивнішим. PyCharm підтримує редагування коду, навігацію, тестування та налагодження, що робить його всебічним інструментом для розробників Python. Однією з головних переваг PyCharm є його інтелектуальний редактор коду, який забезпечує автодоповнення, перевірку синтаксису в режимі реального часу, рефакторинг і аналіз коду. Це допомагає розробникам швидше писати якісний код і виявляти помилки ще до запуску програми.

PyCharm підтримує роботу з фреймворками, такими як Django, Flask та Pyramid, що робить його ідеальним для веб-розробки. Вбудовані інструменти для роботи з базами даних дозволяють розробникам переглядати і редагувати дані безпосередньо з IDE. PyCharm також підтримує інтеграцію з системами контролю версій, такими як Git, Mercurial та SVN, що спрощує управління кодом і співпрацю в команді. PyCharm забезпечує можливість налаштування середовища розробки під потреби конкретного проекту. Вона включає підтримку віртуальних середовищ, що дозволяє легко керувати залежностями проекту. Крім того, PyCharm пропонує

вбудований інтерпретатор Python та підтримку Jupyter Notebook, що робить його корисним інструментом для аналізу даних та наукових досліджень.

PyCharm має розширені можливості налагодження, включаючи інтерактивний налагоджувач, що дозволяє встановлювати точки зупинки, переглядати змінні та виконувати код крок за кроком. Це значно полегшує процес виявлення та виправлення помилок у програмі. Переваги PyCharm включають інтелектуальний редактор коду, що забезпечує автодоповнення, перевірку синтаксису в режимі реального часу та рефакторинг. Підтримка різних фреймворків і бібліотек робить його універсальним інструментом для веб-розробки та аналізу даних. Інтеграція з системами контролю версій та вбудовані інструменти для роботи з базами даних спрощують управління проектами та співпрацю в команді. Розширені можливості налагодження дозволяють швидко знаходити і виправляти помилки.

Проте, PyCharm має і свої недоліки. Одним з них є високе споживання ресурсів, що може вплинути на продуктивність системи, особливо на старих або менш потужних комп'ютерах. Крім того, повна версія PyCharm Professional є платною, хоча існує безкоштовна версія Community Edition з обмеженими функціональними можливостями. PyCharm широко використовується в різних галузях, включаючи веб-розробку, аналіз даних, машинне навчання та наукові дослідження. Його потужні інструменти та функції роблять його популярним вибором серед розробників Python, які цінують ефективність і зручність у роботі.

Visual Studio Code (VS Code) – це потужний, безкоштовний редактор коду з відкритим вихідним кодом, розроблений компанією Microsoft. Він підтримує широкий спектр мов програмування та надає розробникам безліч інструментів і функцій для ефективної роботи з кодом.

VS Code відзначається своєю швидкістю і легкістю, що робить його ідеальним для щоденного використання. Він має інтуїтивно зрозумілий інтерфейс і підтримує численні розширення, що дозволяє налаштовувати редактор під конкретні потреби розробника. Однією з головних переваг VS Code є його

інтегрований термінал, який дозволяє виконувати команди безпосередньо в редакторі, що спрощує процес розробки.

Visual Studio Code підтримує функції автодоповнення коду, рефакторинг, синтаксичний аналіз в режимі реального часу та інтерактивну налагоджувальну систему, що допомагає розробникам швидко знаходити і виправляти помилки. Крім того, він має вбудовану підтримку систем контролю версій, таких як Git, що спрощує управління кодом і співпрацю в команді.

Редактор також підтримує розширення для різних мов програмування, фреймворків і платформ, таких як Python, JavaScript, TypeScript, C++, Java, Go, .NET, Node.js та багато інших. Це робить VS Code універсальним інструментом для розробки різних типів програмного забезпечення.

Однією з ключових характеристик VS Code є його здатність інтегруватися з різними сервісами та інструментами через розширення. Розробники можуть налаштовувати редактор під свої потреби, додаючи розширення для підтримки специфічних мов, інструментів тестування, візуалізації даних, DevOps та інших аспектів розробки програмного забезпечення.

VS Code забезпечує високу продуктивність завдяки своїй архітектурі, побудованій на основі Electron, що дозволяє використовувати його як настільний додаток на різних операційних системах, включаючи Windows, macOS та Linux. Це робить його доступним для широкого кола розробників, незалежно від платформи, яку вони використовують.

Visual Studio Code має активну спільноту користувачів і розробників, які постійно вдосконалюють редактор, додаючи нові функції та виправляючи помилки. Це забезпечує постійний розвиток редактора і робить його одним з найбільш популярних інструментів для розробників у всьому світі.

4 РОЗРОБКА ТА ТЕСТУВАННЯ ДОДАТКУ ДЛЯ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ТА АНАЛІЗУ ВИТРАТ В ДОМОГОСПОДАРСТВАХ

4.1 Опис етапів розробки

Розробка додатку для автоматизованого управління та аналізу витрат в домогосподарствах включає кілька ключових етапів, що забезпечують аналіз, планування, розробку, тестування.

Спочатку здійснюється збір вимог, визначаються основні функціональні та нефункціональні вимоги до додатку.

Потім проводиться детальний аналіз ринку існуючих аналогів та визначення функціональності, яку має виконувати веб-додаток. Аналізуються додатки, такі як Personal Capital, PocketGuard та YNAB, встановлюються основні функції. Також визначаються відмінні риси додатку, які можуть спонукати користувача обрати для користування саме розроблений додаток.

Проектування системи: на цьому етапі розробляється архітектура веб-додатку для автоматизованого управління та аналізу витрат в домогосподарстві. Визначаються основні компоненти, включаючи фронтенд, бекенд, базу даних. Створюється дизайн бази даних для зберігання транзакцій, категорій, користувачів та інших необхідних даних. Проектується логіка обробки даних та взаємодії між компонентами.

Реалізація: на цьому етапі розпочинається написання коду. Для його написання були використані наступні технології:

Python: для реалізації бекенду, обробки даних та інтеграції з базою даних.

JavaScript: для забезпечення інтерактивності та маніпуляцій з даними на фронтенді.

HTML/CSS: для створення користувацького інтерфейсу.

SQLite: для створення та управління базою даних.

Реалізуються основні функції додатку, такі як додавання та видалення транзакцій, створення звітів та діаграм, інтеграція з базою даних та створення API для взаємодії з фронтендом.

Тестування: після завершення реалізації проводиться тестування веб-додатку. Тестування включає мануальне тестування для виявлення дефектів, помилок та недоліків. Мануальне тестування дозволяє перевірити, чи відповідає додаток функціональним вимогам та очікуванням кінцевих користувачів.

Ці етапи забезпечують систематичний підхід до розробки програмного забезпечення, що дозволяє створити ефективну, надійну та зручну систему для автоматизованого управління та аналізу витрат в домогосподарствах.

4.2 Моделювання додатку

Моделювання додатку для автоматизованого управління та аналізу витрат в домогосподарствах є важливим етапом у процесі розробки програмного забезпечення, оскільки воно дозволяє створити детальну уяву про структуру, функціональність і взаємодію компонентів системи. Цей процес включає визначення основних вимог до системи, розробку архітектури, проектування користувацького інтерфейсу і планування бази даних.

На початковому етапі моделювання необхідно визначити, які функції повинні бути реалізовані в додатку. Це можуть бути такі функції як введення та категоризація витрат, встановлення бюджетів, генерування графіків. Важливо також враховувати нефункціональні вимоги, продуктивність та зручність використання.

Після збору вимог розробляється архітектура додатку. Це включає визначення основних компонентів системи і їх взаємодії. Архітектура складається з фронтенду, який відповідає за користувацький інтерфейс, з бекенду, який обробляє бізнес-логіку і взаємодію з базою даних, та бази даних, яка зберігає всі фінансові дані користувачів. Важливо забезпечити, щоб всі компоненти системи були добре інтегровані і могли ефективно взаємодіяти один з одним.

Проектування користувацького інтерфейсу є ключовою частиною моделювання додатку. Інтерфейс повинен бути інтуїтивно зрозумілим і зручним для користувачів, щоб вони могли легко вводити дані, переглядати графічні дані про свої витрати.

Планування бази даних включає визначення структури даних, які будуть зберігатися в системі. Це може включати таблиці для зберігання інформації про користувачів, витрати, категорії витрат, бюджети та інші фінансові дані. Важливо розробити ефективну структуру бази даних, яка дозволить швидко отримувати необхідну інформацію і підтримувати цілісність даних.

Моделювання поведінки додатку включає визначення процесів і сценаріїв використання системи, що дозволяє зрозуміти, як користувачі будуть взаємодіяти з додатком і які кроки необхідні для виконання різних завдань.

Завершальним етапом моделювання є тестування. Це дозволяє виявити можливі проблеми і недоліки в дизайні системи до початку її реалізації.

Таким чином, моделювання додатку для автоматизованого управління та аналізу витрат в домогосподарствах є комплексним процесом, що включає збір вимог, розробку архітектури, проектування користувацького інтерфейсу, планування бази даних і тестування. Це дозволяє створити детальну уяву про систему і забезпечити її ефективно і зручне використання.

4.3 Діаграма варіантів використання

Діаграма варіантів використання для додатку управління та аналізу витрат в домогосподарствах являє собою візуальне представлення різних функцій та сценаріїв взаємодії користувачів із системою. Вона допомагає зрозуміти, які функції будуть доступні користувачам та як вони взаємодіють із системою для досягнення своїх цілей. На діаграмі варіантів використання відображаються актори (користувачі або інші системи) і варіанти використання (функції системи).

Основним актором на діаграмі є користувач додатку, який може виконувати різні дії для управління своїми фінансами. Після реєстрації користувач входить в систему, використовуючи свої облікові дані.

Після входу користувач отримує доступ до основних функцій додатку. Однією з основних функцій є додавання витрат, де користувач вводить інформацію про свої витрати, включаючи суму, категорію витрат і дату. Інша важлива функція – це перегляд бюджету, де користувач може створювати і переглядати свої бюджети, встановлюючи ліміти витрат для різних категорій.

Додаток також надає можливість переглядати графіки витрат. Цей варіант використання дозволяє користувачу генерувати графіки витрат за певний період часу, щоб проаналізувати свої витрати і виявити можливі області для економії. Створені графіки наочно показують структуру витрат.

Таким чином, діаграма варіантів використання для додатку управління та аналізу витрат в домогосподарствах включає основні функції, такі як реєстрація користувачів, вхід у систему, додавання витрат, перегляд бюджету, генерування графіків витрат. Це візуальне представлення допомагає зрозуміти, які функції будуть доступні користувачам і як вони зможуть взаємодіяти з системою для ефективного управління своїми фінансами. На рисунку 4.1 наведено діаграму варіантів використання.



Рис. 4.1 Діаграма варіантів використання

4.4 Проектування інтерфейсу

Інтерфейс визначає, як користувачі будуть взаємодіяти з системою і наскільки зручно і ефективно вони зможуть виконувати свої фінансові завдання.

Проектування інтерфейсу користувача (UI) є ключовим етапом у створенні веб-програми. Інтерфейс користувача має бути інтуїтивно зрозумілим, зручним та естетично привабливим, щоб забезпечити ефективну взаємодію користувачів з програмою.

Головна сторінка містить назву веб-додатку з логотипом та навігаційне меню, що надає доступ до функцій програми, таких як реєстрація та вхід до системи. Нижня частина сторінки містить контактну інформацію, посилання на політику конфіденційності та умови користування. На рисунку 4.2 наведено головну сторінку.

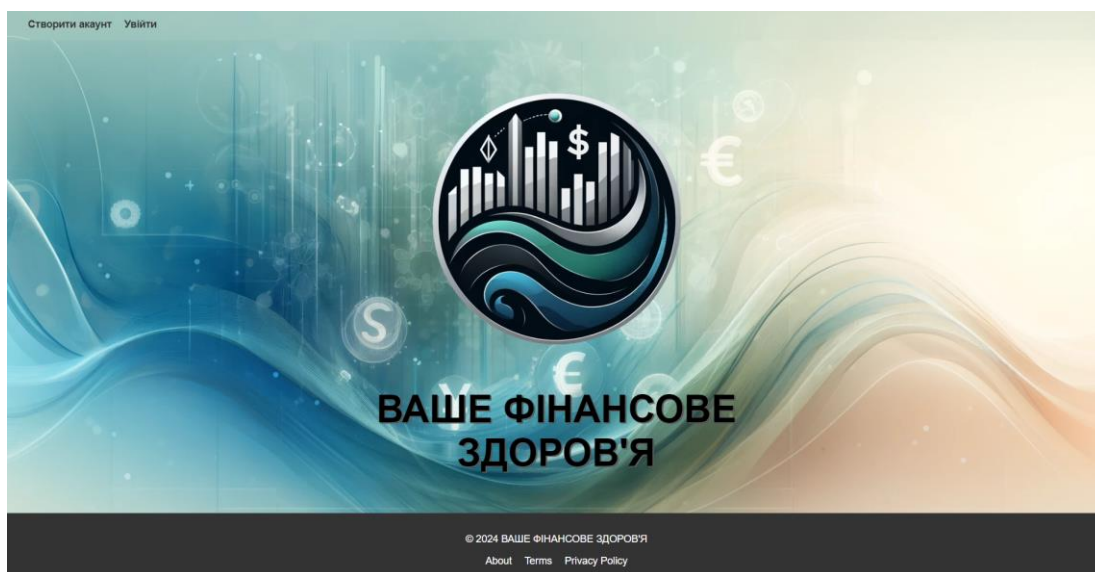


Рис. 4.2 Головна сторінка

Сторінка входу забезпечує простий та зручний спосіб авторизації користувачів. Вона включає форму входу з полями для введення імені користувача та пароля, а також базову валідацію введених даних. Для користувачів, які ще не

мають облікового запису, передбачено посилання на сторінку реєстрації. На рисунку 4.3 наведено сторінку авторизації.

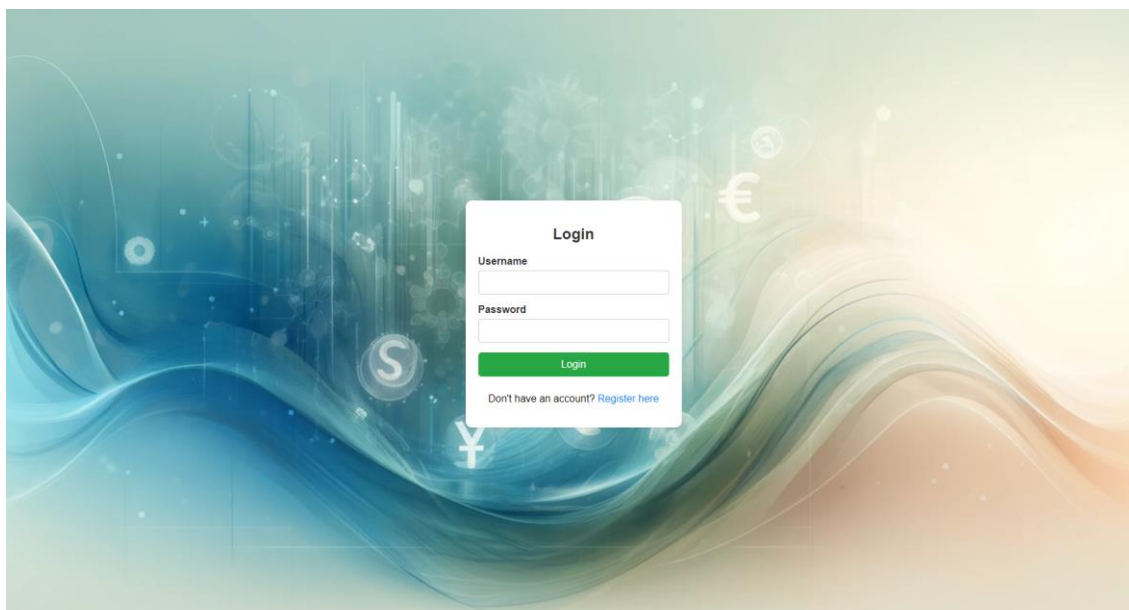


Рис. 4.3 Сторінка входу

Сторінка реєстрації дозволяє новим користувачам створити обліковий запис. Вона містить форму з полями для введення імені користувача, електронної пошти, пароля та його підтвердження. Реалізована валідація відповідності паролів для забезпечення коректності введених даних. Після успішної реєстрації користувач перенаправляється на сторінку входу. На рисунку 4.4 наведено сторінку реєстрації.

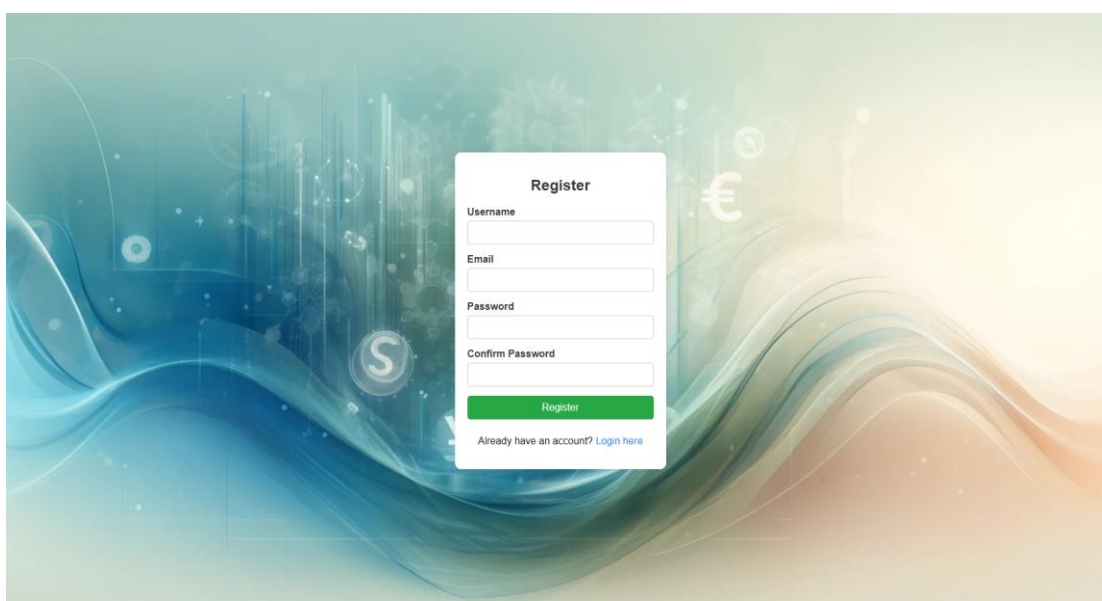
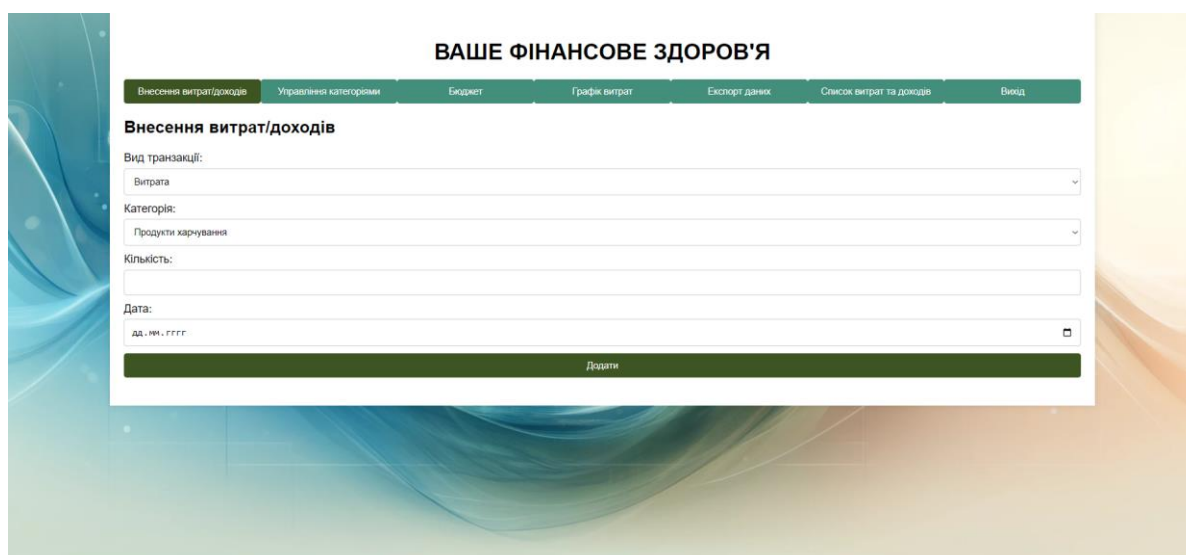


Рис. 4.4 Сторінка реєстрації

Головна сторінка після входу до системи забезпечує доступ до основних функцій програми через навігацію по вкладках. Кожна вкладка відповідає за окрему функцію програми, включаючи внесення витрат та доходів, управління категоріями, створення бюджету, генерування графіку витрат, експорт даних та перегляд списку транзакцій. Це дозволяє користувачам легко знаходити та використовувати необхідні функції.

Форма для внесення витрат та доходів дозволяє користувачам додавати нові транзакції з вибором типу транзакції (дохід чи витрата), категорії, суми та дати. На рисунку 4.5 наведено вкладку внесення витрат та доходів.



The screenshot displays a web application interface titled "ВАШЕ ФІНАНСОВЕ ЗДОРОВ'Я". At the top, there is a navigation bar with several tabs: "Внесення витрат/доходів", "Управління категоріями", "Бюджет", "Графік витрат", "Експорт даних", "Список витрат та доходів", and "Вихід". The "Внесення витрат/доходів" tab is currently selected. Below the navigation bar, the form is titled "Внесення витрат/доходів". It contains the following fields: "Вид транзакції:" with a dropdown menu showing "Витрата"; "Категорія:" with a dropdown menu showing "Продукти харчування"; "Кількість:" with a text input field; "Дата:" with a date picker showing "дд.мм.рррр"; and a "Додати" button at the bottom.

Рис. 4.5 Внесення витрат та доходів

Управління категоріями забезпечує можливість додавання нових категорій та їх видалення. На рисунку 4.6 наведено вкладку управління категоріями.

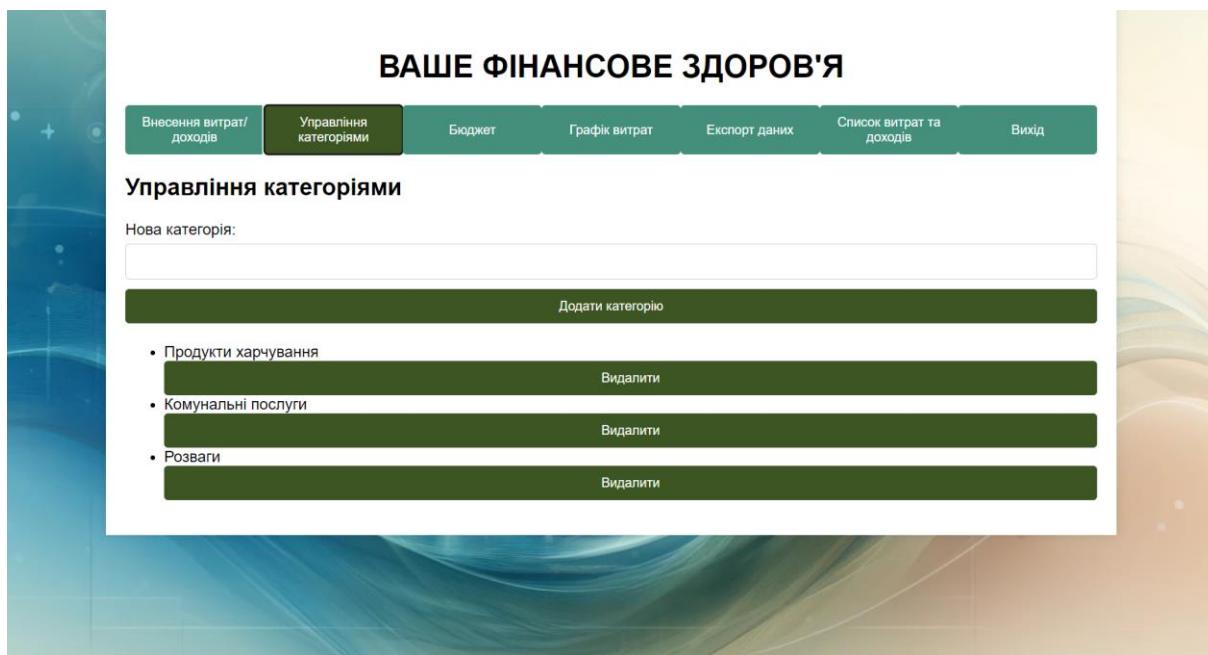


Рис. 4.6 Управління категоріями

Функція створення бюджету дозволяє встановити загальний бюджет та бюджети по окремих категоріях на певний період часу. На рисунку 4.7 наведено вкладку зі створеним бюджетом.

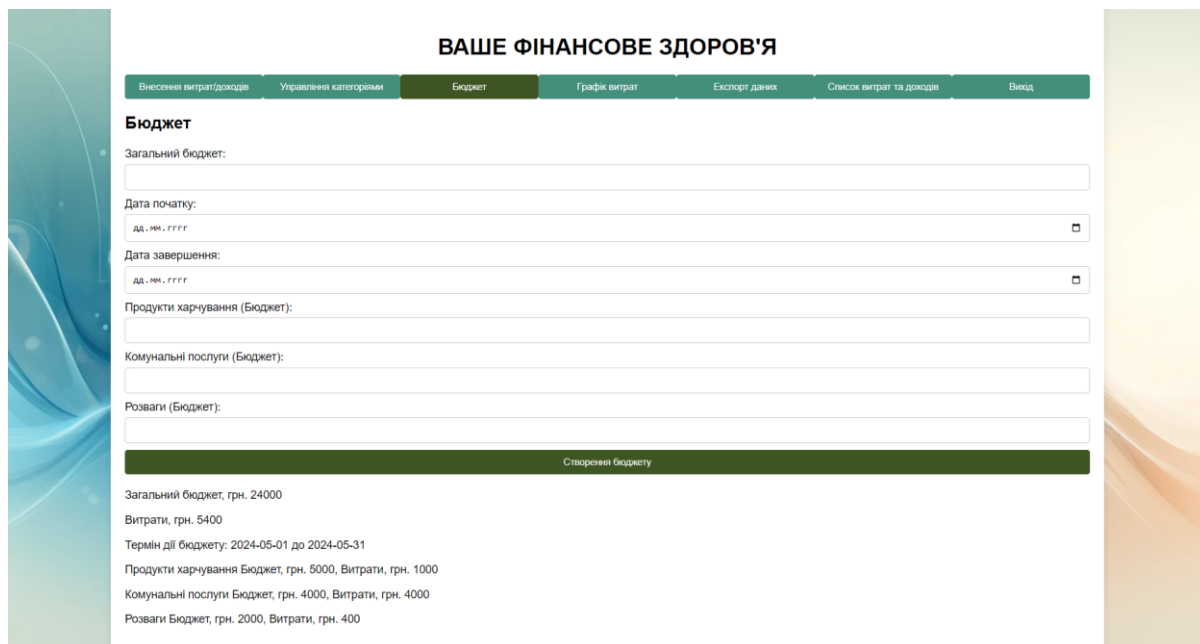


Рис. 4.7 Бюджет

Вкладка з графічними звітами надає візуалізацію витрат за допомогою графіків. На рисунку 4.8 наведено вкладку зі згенерованим графіком витрат.

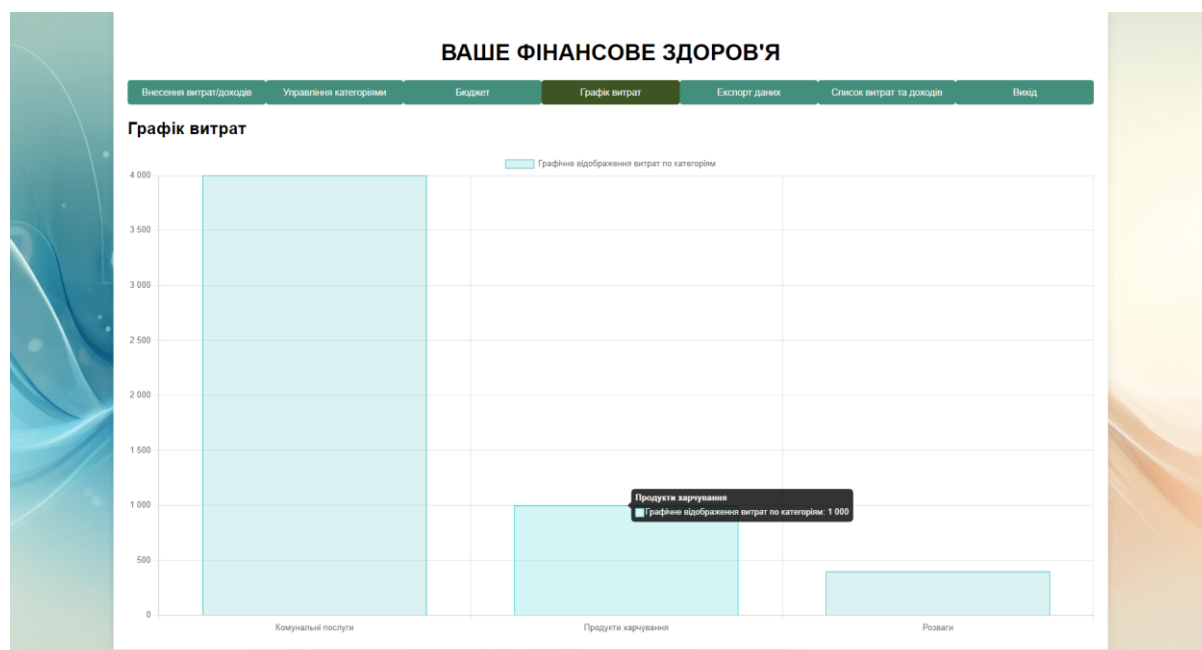


Рис. 4.8 Графік витрат

Функція експорту даних дозволяє користувачам зберігати свої транзакції у форматі CSV, що забезпечує зручність у подальшому аналізі. На рисунку 4.9 наведено вкладку експорту даних.



Рис. 4.9 Експорт даних

Перегляд списку витрат та доходів надає користувачам можливість швидко ознайомитися з усіма внесеними транзакціями. На рисунку 4.10 наведено вкладку зі списком витрат та доходів.

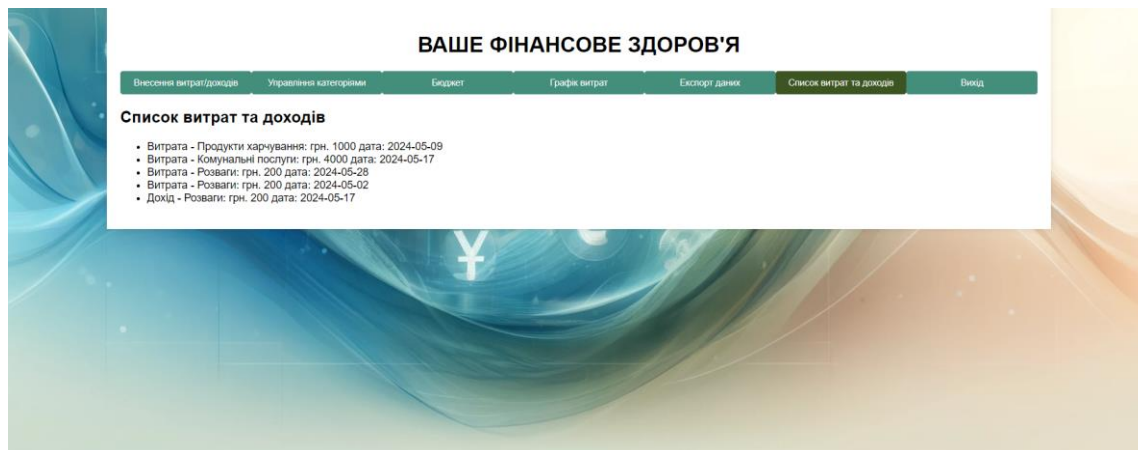


Рис. 4.10 Список витрат та доходів

Під час проектування інтерфейсу важливо враховувати мобільність користувачів. Додаток повинен бути адаптивним і добре працювати як на десктопних, так і на мобільних пристроях. Це дозволить користувачам керувати своїми фінансами в будь-який час і в будь-якому місці.

4.5 Діаграма діяльності

Діаграма діяльності є графічним інструментом, що використовується в UML (Unified Modeling Language) для моделювання послідовності дій або робочих процесів у системі. Вона надає можливість візуально представити процеси, які виконуються в системі, а також потоки управління і даних між різними частинами системи. Основна мета діаграми діяльності - це відображення того, як виконуються різні дії і як вони взаємодіють одна з одною.

Діаграма діяльності використовує різні спеціалізовані елементи.

Наприклад, дія, представлена прямокутником із заокругленими кутами, описує конкретну операцію або крок у процесі. Потоки управління, які

показуються стрілками, визначають послідовність виконання дій. Початковий вузол, заповнене чорне коло, вказує на початкову точку процесу.

Діаграми діяльності використовуються для моделювання різних типів робочих процесів, включаючи бізнес-процеси, алгоритми, сценарії використання та інші динамічні аспекти системи. Вони забезпечують всебічний огляд процесів і допомагають розробникам та аналітикам зрозуміти, як різні частини системи взаємодіють і як виконуються окремі дії. Цей інструмент є важливим засобом для аналізу, проектування і документування систем, допомагаючи забезпечити їх ефективність і надійність. На рисунку 4.11 наведено діаграму діяльності.

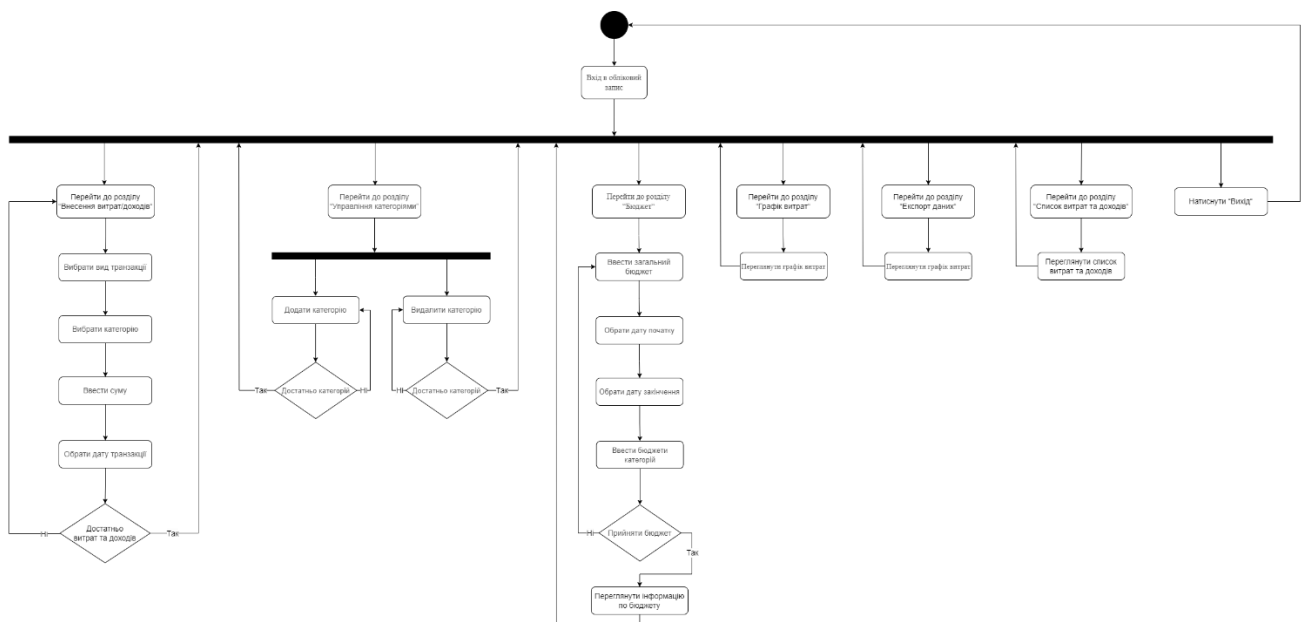


Рис. 4.11 Діаграма діяльності

4.6 Діаграма класів

Діаграма класів показує основні класи, їхні атрибути, методи та взаємозв'язки між ними. Ця діаграма допомагає зрозуміти структуру програмного забезпечення і його компоненти.

Діаграма класів для додатку управління та аналізу витрат у домогосподарствах включає основні класи, такі як Користувач, Транзакція, Бюджет, Категорія.

Основний клас на діаграмі – це клас Користувач. Клас Користувач містить атрибути, такі як ім'я, електронна пошта, пароль. Методи цього класу включають функції для реєстрації та входу в систему. Клас Користувач також має асоціації з іншими класами, такими як Транзакція, Бюджет та Категорія.

Клас Транзакція представляє інформацію про окремі транзакції користувача. Цей клас містить атрибути, такі як вид транзакції, сума витрат, дата, категорія. Методи класу включають додавання нових витрат та доходів. Клас Транзакція має асоціацію з класом Користувач, оскільки кожна витрата належить певному користувачу, також має асоціацію з класом Категорія, оскільки кожна транзакція має категорію.

Клас Бюджет відповідає за управління фінансовими планами користувача. Він містить атрибути, такі як категорії витрат, ліміти бюджету та період дії бюджету. Методи класу включають створення нового бюджету, оновлення існуючого бюджету. Клас Бюджет також асоційований з класами Користувач та Категорія.

Також на діаграмі присутній клас Категорія, який представляє різні типи витрат, такі як харчування, транспорт, житло тощо. Клас Категорія містить атрибут, такий як назва категорії. Методи класу включають створення нових категорій та видалення існуючих категорій. Клас Категорія має асоціації з класами Транзакція та Бюджет, оскільки витрати і бюджети пов'язані з певними категоріями. На рисунку 4.12 наведено діаграму класів. На рисунках 4.13 – 4.16 виведені фрагменти коду деяких класів та їх функцій.

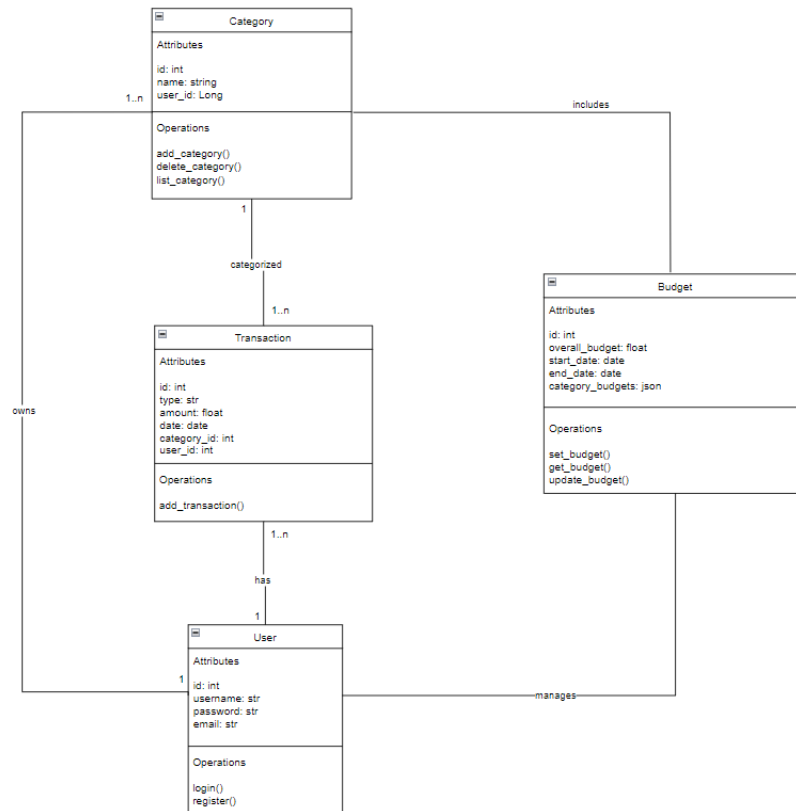


Рис. 4.12 Діаграма класів

```

class Transaction(db.Model):
    __tablename__ = 'transactions'
    transaction_id = db.Column(db.Integer, primary_key=True)
    user_id = db.Column(db.Integer, db.ForeignKey('users.user_id'), nullable=False)
    category = db.Column(db.String(80), nullable=False)
    amount = db.Column(db.Float, nullable=False)
    date = db.Column(db.Date, nullable=False)
    transaction_type = db.Column(db.String(80), nullable=False)
  
```

Рис. 4.13 Фрагмент коду класу Transaction

```

@staticmethod
def add_transaction(user_id, category, amount, date, transaction_type):
    transaction = Transaction(
        user_id=user_id,
        category=category,
        amount=amount,
        date=datetime.strptime(date, '%Y-%m-%d'),
        transaction_type=transaction_type
    )
    db.session.add(transaction)
    db.session.commit()
    return transaction
  
```

Рис. 4.14 Фрагмент коду функції класу Transaction

```
class Category(db.Model):
    __tablename__ = 'categories'
    category_id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(80), unique=True, nullable=False)
    user_id = db.Column(db.Integer, db.ForeignKey('users.user_id'), nullable=False)
```

Рис. 4.15 Фрагмент коду класу Category

```
@staticmethod
def add_category(name, user_id):
    if not Category.query.filter_by(name=name, user_id=user_id).first():
        category = Category(name=name, user_id=user_id)
        db.session.add(category)
        db.session.commit()
        return True
    return False

@staticmethod
def delete_category(name):
    category = Category.query.filter_by(name=name).first()
    if category:
        db.session.delete(category)
        db.session.commit()
        return True
    return False

@staticmethod
def list_categories():
    return [category.to_dict() for category in Category.query.all()]
```

Рис. 4.16 Фрагмент коду функцій класу Category

4.7 Схема баз даних

Схема бази даних для додатку управління та аналізу витрат у домогосподарствах являє собою структуру таблиць з відображенням їхніх взаємозв'язків, що забезпечують зберігання та організацію даних у системі. Основна мета такої схеми – забезпечити ефективне зберігання даних, швидкий доступ до них і підтримку цілісності інформації.

Схема бази даних для веб-додатку управління та аналізу витрат у домогосподарствах включає кілька ключових таблиць: Користувачі, Транзакції, Категорії, Бюджет, Бюджети за категоріями. Ці таблиці взаємопов'язані, що забезпечує ефективне зберігання, організацію та доступ до даних, необхідних для функціонування системи.

У схемі бази даних першочерговою таблицею є Користувачі. Ця таблиця містить інформацію про користувачів додатку, включаючи їхні ідентифікатори,

імена, електронні адреси та паролі. Кожен запис у таблиці Користувачі відповідає одному користувачу системи.

Важливою таблицею є Транзакції. Вона зберігає дані про окремі транзакції користувачів, включаючи ідентифікатор транзакцій, ідентифікатор користувачів, вид транзакції, суму транзакції, дату транзакції, категорію транзакції. Таблиця Транзакції має зв'язок із таблицею Користувачі через ідентифікатор користувача, що дозволяє зв'язати кожен витрату чи дохід з певним користувачем.

Таблиця Категорії зберігає інформацію про різні категорії витрат та доходів. Вона містить ідентифікатори категорій, назви категорій. Таблиця Категорії пов'язана з таблицями Користувачі, Транзакції, Бюджети за категоріями, що дозволяє категоризувати кожен транзакцію, пов'язати кожен категорію з певним користувачем та створювати бюджети за категоріями.

Таблиця Бюджет зберігає дані про фінансові плани користувачів. Вона включає ідентифікатори бюджетів, ідентифікатори користувачів, ліміт бюджету та період дії бюджету. Таблиця Бюджет має зв'язки з таблицею Користувачі, що дозволяє визначити, які бюджети належать певним користувачам.

Таблиця Бюджети за категоріями зберігає дані про фінансові плани користувачів за категоріями. Вона включає ідентифікатори бюджетів, ідентифікатори користувачів, ліміт бюджету та категорії. Таблиця Бюджети за категоріями має зв'язки з таблицями Користувачі, Категорії, що дозволяє визначити, які бюджети належать певним користувачам та створювати бюджети за категоріями. На рисунку 4.17 наведено схему баз даних.

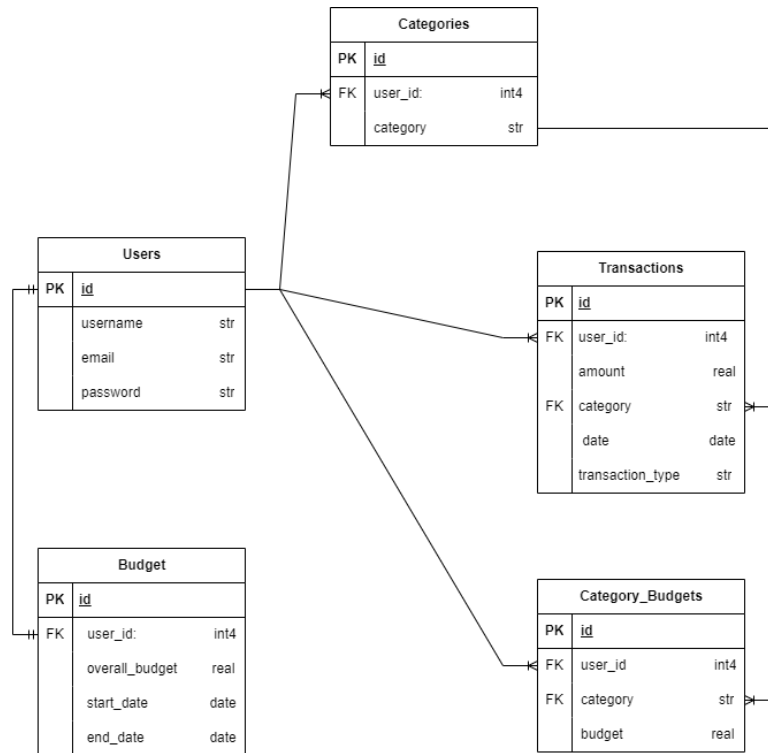


Рис. 4.17 Схема баз даних

4.8 Мапа сайту

Мапа сайту веб-додатку для управління та аналізу витрат у домогосподарствах – це структуроване представлення всіх основних сторінок і розділів додатку, що забезпечує зручну навігацію для користувачів. Вона включає головну сторінку, сторінку авторизації, розділи «Внесення витрат/доходів», «Управління категоріями», «Бюджет», «Графік витрат», «Експорт даних», «Список витрат та доходів» та кнопку «Вихід». Мапа сайту допомагає зрозуміти, як організовані різні функції та сервіси додатку, і які кроки потрібно виконати для доступу до них.

Головна сторінка додатку є точкою входу. З головної сторінки користувачі можуть переходити до реєстрації та входу.

Сторінка авторизації включає функції реєстрації та входу. Тут користувачі можуть створювати новий обліковий запис та входити в систему.

Розділ "Внесення витрат/доходів" дозволяє користувачам додавати, витрати та доходи. Тут вони можуть вводити інформацію про свої транзакції, включаючи вид транзакції, суму, категорію, дату.

Розділ «Управління категоріями» дозволяє додавати та видаляти категорії.

Розділ "Бюджет" призначений для створення і управління бюджетами. Користувачі можуть налаштовувати бюджети для різних категорій витрат, встановлювати ліміти і контролювати їх виконання. У цьому розділі відображаються загальні витрати біля значення загального ліміту, та загальні витрати по категоріям біля лімітів по цим категоріям, для легшого контролю виконання цих бюджетів.

Розділ "Графік витрат" забезпечує доступ до аналітичних графіків про витрати користувача. Користувачі можуть переглядати результати у вигляді згенерованих графіків. Цей розділ допомагає користувачам аналізувати свої витрати і приймати обґрунтовані рішення щодо економії коштів.

Розділ «Експорт даних» дозволяє користувачу скачати файл з інформацією про витрати та доходи.

Кнопка «Вихід» дозволяє користувачу вийти з облікового запису та переносить його до головної сторінки. На рисунку 4.18 наведено мапу сайту.

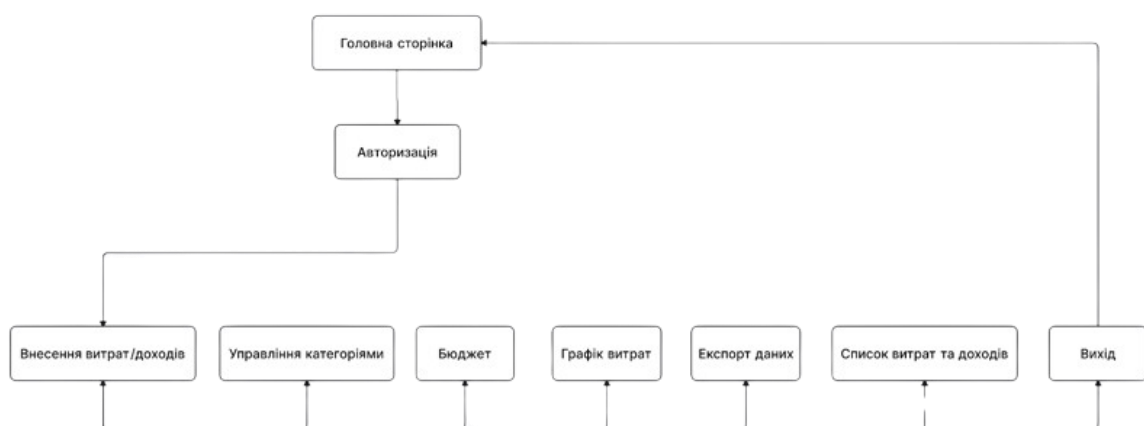


Рис. 4.18 Мапа сайту

4.9 Тестування програмного забезпечення

Тестування додатку для управління та аналізу витрат у домогосподарствах – це етап у процесі розробки, який забезпечує якість, надійність та відповідність системи вимогам користувачів. Метою тестування є виявлення помилок, дефектів та недоліків у додатку, щоб гарантувати його безперебійну роботу в реальних умовах.

Для веб-додатку "МОЄ ФІНАНСОВЕ ЗДОРОВ'Я" використовувалось мануальне тестування з наступними тестовими сценаріями:

Вхід в обліковий запис або реєстрація нового аккаунту

1. Перевірка коректності обробки правильних та неправильних облікових даних.

2. Тестування реєстрації нового аккаунту, перевірка правильності заповнення полів та валідації введених даних.

Ведення власного фінансового обліку

1. Тестування функцій внесення доходів та витрат, переконання в коректному записі та збереженні даних у системі.

2. Випробування різних варіантів введення даних, включаючи вид транзакцій, суми, категорії та дати, для перевірки правильності обробки цих даних.

Управління категоріями

1. Тестування функцій внесення та видалення категорій.

2. Тестування функції перегляду доступних категорій.

Управління бюджетом

1. Перевірка можливості встановлення загального бюджету, бюджетів за категоріями та можливість встановлення граничних дат цих бюджетів.

2. Тестування функції перегляду інформації про загальний бюджет, загальні витрати, граничні дати, бюджети за категоріями та витрати в межах бюджетів за категоріями.

Експорт даних

1. Перевірка можливості експорту даних у CSV форматі.

2. Тестування коректності експорту даних з урахуванням введених даних.

Список витрат та доходів

1. Перевірка правильності відображення списку витрат та доходів.

Метод мануального тестування був обраний та використаний, бо він дозволяє, під час тестування, активно взаємодіяти з системою та динамічно перевіряти її реакцію на різні вхідні дані та сценарії. Це забезпечує більш гнучкий та деталізований підхід до тестування, що сприяє виявленню потенційних проблем.

ВИСНОВКИ

В результаті виконання дипломної роботи було розроблено веб-додаток для підтримки процесів автоматизованого управління та аналізу витрат в домогосподарствах мовою Python.

Актуальність теми зумовлена ускладненням економічних умов, що призводить до зростання потреби у фінансовій грамотності та безпеці домогосподарств; труднощістю та часу, які витрачаються на ведення обліку витрат, що виявляє недоліки традиційних методів управління та аналізу витрат.

В ході вирішення поставлених задач було:

1. Проаналізовано процес управління та аналізу витрат в домогосподарствах.
2. Проаналізовані засоби, які використовуються для ручного та автоматизованого управління та аналізу витрат в домогосподарствах.
3. Визначені переваги та недоліки існуючих програмних засобів, результати проведеного аналізу були використані при розробці застосунку для управління та аналізу витрат в домогосподарствах. Розглянуті такі засоби як PocketGuard, Personal Capital та YNAB.
4. Розроблено функціональні та нефункціональні вимоги до застосунку для автоматизованого управління та аналізу витрат.
5. Спроектовано архітектуру, визначено класи для створення застосунку для автоматизованого управління та аналізу витрат в домогосподарствах, побудовані: діаграма варіантів використання, діаграма діяльності, діаграма класів.
6. Розроблено програмне забезпечення для автоматизованого управління та аналізу витрат в домогосподарствах з використанням мов Python, JavaScript; фреймворка Flask; бібліотек: Pandas, SQLAlchemy; СУБД SQLite; мови розмітки HTML; середовищ розробки PyCharm, Visual Studio Code; мови стилю CSS.
7. Проведено ручне тестування застосунку для автоматизованого управління та аналізу витрат в домогосподарствах.

Розробка програмного забезпечення з інтуїтивно зрозумілими функціями, що нагадують звичайні методи ведення аналізу витрат, може значно спонукати

споживачів переходити на автоматизоване управління ними. Коли користувачі бачать знайомі елементи та функції, вони легше адаптуються до нової технології.

В свою чергу, перехід на автоматизоване управління та аналіз витрат, допоможе домогосподарствам ефективно управляти своїми фінансами, аналізувати витрати та планувати свої фінансові ресурси.

Дипломна робота пройшла апробацію на трьох конференціях: Дипломна робота пройшла апробацію на двох конференціях:

1. Казьмірчук О.Г. Засоби мови програмування Python для захисту інформації. XI Всеукраїнська науково-практична конференція молодих учених «Інформаційні технології – 2024» (ІТ-2024), 17 травня 2024 р., Київ, Київський столичний університет імені Бориса Грінченка. Збірник тез. К.: Київський столичний університет імені Бориса Грінченка, 2024, С. 238-240,

2. Казьмірчук О.Г., Садовенко В.С. Розробка програмного забезпечення аналізу витрат в домогосподарствах мовою Python. Всеукраїнська науково-технічна конференція “Застосування програмного забезпечення в інфокомунікаційних технологіях”, 24 травня 2024 р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2024, С. 88-90

3. Казьмірчук О.Г., Садовенко В.С. Методи інтелектуального аналізу даних та засоби їх реалізації мовою Python у фінансовому плануванні домогосподарств. IV Всеукраїнська науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті», 15 травня 2024 р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2024, С. 56-57.

ПЕРЕЛІК ПОСИЛАНЬ

1. Поліщук О.В. Вміння ефективно управляти витратами та отримувати від них максимальний результат є вирішальним чинником стабільності функціонування підприємства у ринкових умовах господарювання. *Економіка. Фінанси. Право*. 2020. № 4/1. С. 27-34.
2. Назарова О.Ю., Чуприна О.А., Чатченко Т.В. Доходи та витрати населення: аналітично-статистичний аспект. *Економіка та суспільство*. 2021. №31. <https://economyandsociety.in.ua/index.php/journal/article/view/680/653>
3. Грабар О. І., Остроухов М. С., Постова С. А. Аналіз сучасних інформаційних технологій автоматизації обліку підприємств. *Вчені записки ТНУ імені В. І. Вернадського*. 2020. № 1. Т. 31(70). Ч. 1. С. 66–71.
4. Лось В. О., Максишко Н. К., Столярчук І. А. Автоматизація обліку грошових коштів домогосподарств на базі «1С: підприємство 8». *Вісник Запорізького національного університету*. 2015. № 4 (28). С. 50–58.
5. Лукановська І.Р. Особливості інформаційного забезпечення автоматизації аналізу витрат. Електронний репозитарій ДВНЗ "УжНУ". 2018. <http://dspace.wunu.edu.ua/bitstream/316497/29617/1/244.PDF>.
6. Лошенюк О.В. Автоматизація управління бізнес-процесами підприємства як основа гнучкості його діяльності: характеристика систем управління. *Економіка та суспільство*. 2022. №45. <https://doi.org/10.32782/2524-0072/2022-45-46>.
7. Кізіма Т. Управління фінансами домашніх господарств: теоретикопрагматичні аспекти / Тетяна Кізіма // *Світ фінансів*. 2011. №4 (21). С. 22–33.
8. М.І. Колосінська, С.І. Белей, С.В. Бойда. Економічна теорія: навчальний посібник. Чернівці: Чернівецький національний університет. 2021. 200 с.
9. YNAB. URL:<https://www.ynab.com> (дата звернення 07.03.2024).
10. Python Software Foundation. Python Documentation. URL: <https://docs.python.org/3.9/> (дата звернення 26.03.2024).

11. Pandas. URL:<https://pandas.pydata.org> (дата звернення 27.03.2024).
12. Що таке html? URL:https://css.in.ua/article/shcho-take-css_3 (дата звернення 28.03.2024).
13. Що таке CSS. URL:https://css.in.ua/article/shcho-take-html_10 (дата звернення 28.03.2024).
14. Довідник по JavaScript. URL:<https://css.in.ua/js/objects> (дата звернення 29.03.2024).
15. SQLite. URL:<https://docs.python.org/3/library/sqlite3.html> (дата звернення 01.04.2024).
16. Казьмірчук О.Г. Засоби мови програмування Python для захисту інформації //XI Всеукраїнська науково-практична конференція молодих учених «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ – 2024» (ІТ-2024). Збірник тез. 17.05.2024, Київський столичний університет імені Бориса Грінченка, Київ. К.: Київський столичний університет імені Бориса Грінченка, 2024, С. 238-240.
17. Садовенко В.С., Казьмірчук О.Г. Розробка програмного забезпечення аналізу витрат в домогосподарствах мовою Python //Всеукраїнська науково-технічна конференція “Застосування програмного забезпечення в інфокомунікаційних технологіях”. Збірник тез. 24.05.2024, ДУІКТ, Київ. К.: ДУІКТ, 2024, С. 88-90.

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка програмного забезпечення підтримки процесів автоматизованого управління та аналізу витрат в домогосподарствах мовою Python

Виконав: студент 4 курсу
групи ПД-43
Казьмірчук Олексій Геннадійович
Керівник роботи

К.ф-м.н., доцент, професор кафедри ІПЗ Садовенко Володимир Сергійович

Київ – 2024

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ





- **Мета роботи** – спрощення управління та аналізу витрат в домогосподарствах за допомогою застосування мовою Python.
- **Об'єкт дослідження** – процес управління та аналізу витрат в домогосподарствах.
- **Предмет дослідження** – програмне забезпечення, яке допомагає автоматизувати процеси управління та аналізу витрат в домогосподарствах.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Проаналізувати управління та аналіз витрат в домогосподарствах.
2. Проаналізувати засоби, які можуть бути використані для управління та аналізу витрат в домогосподарствах.
3. Визначити переваги та недоліки існуючих програмних засобів.
4. Розробити функціональні та нефункціональні вимоги до застосунку для автоматизованого управління та аналізу витрат в домогосподарствах.
5. Спроекувати архітектуру, визначити класи для створення застосунку для автоматизованого управління та аналізу витрат в домогосподарствах.
6. Розробити програмне забезпечення для автоматизованого управління та аналізу витрат в домогосподарствах.
7. Провести тестування застосунку для автоматизованого управління та аналізу витрат в домогосподарствах.

3

АНАЛІЗ АНАЛОГІВ

Показник	 PocketGuard	 Personal Capital	 YNAB	 ВАШЕ ФІНАНСОВЕ ЗДОРОВ'Я
Категорії витрат	фіксований список категорій	фіксований список категорій	фіксований список категорій	є можливість додавання власних категорій
Планування бюджету	+	+	+	+
Розподіл бюджету за категоріями	-	-	-	+
Визначення термінів бюджету	термін фіксований (30, 60, 90 днів)	термін фіксований (30, 60, 90 днів)	термін фіксований (30, 60, 90 днів)	є можливість самостійно обрати граничні дати
Графічні звіти витрат	+	-	+	+
Експорт фінансових даних	-	-	-	+

4

ВИМОГИ ДО ДОДАТКУ

Функціональні вимоги до додатку:

1. Можливість реєстрації та авторизації.
2. Можливість введення користувачем доходів та витрат.
3. Можливість додавання та видалення категорій.
4. Можливість створення бюджету.
5. Можливість створення бюджету із зазначенням початкової та кінцевої дати.
6. Можливість генерування графічного звіту витрат.

Нефункціональні вимоги.

1. Локалізація українською мовою.
2. Сумісність з сучасними веб-браузерами (Chrome, Firefox, Safari, Edge, Opera).
3. Підтримка різних розмірів екрану (ПК, планшети, смартфони).

5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



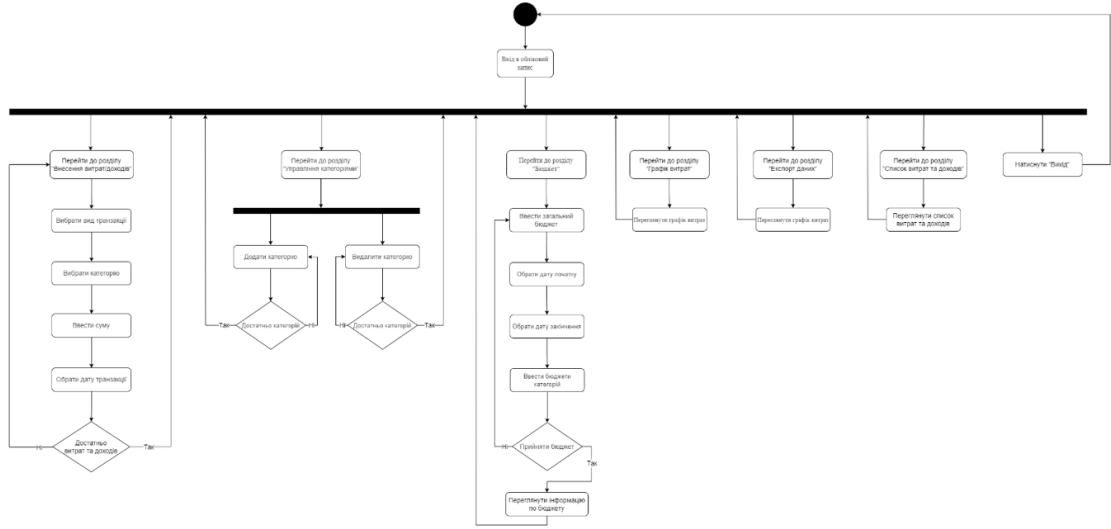
6

Діаграма варіантів використання



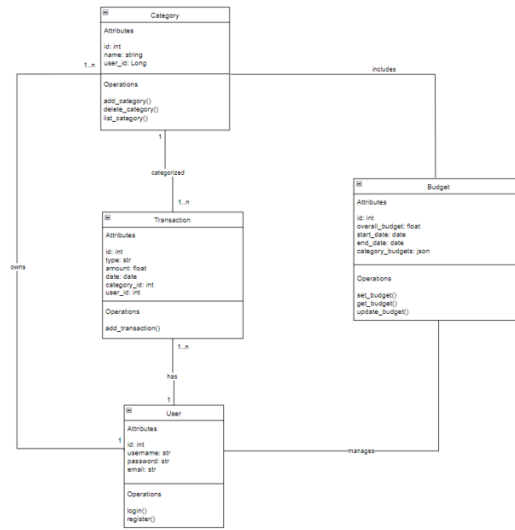
7

Діаграма діяльності



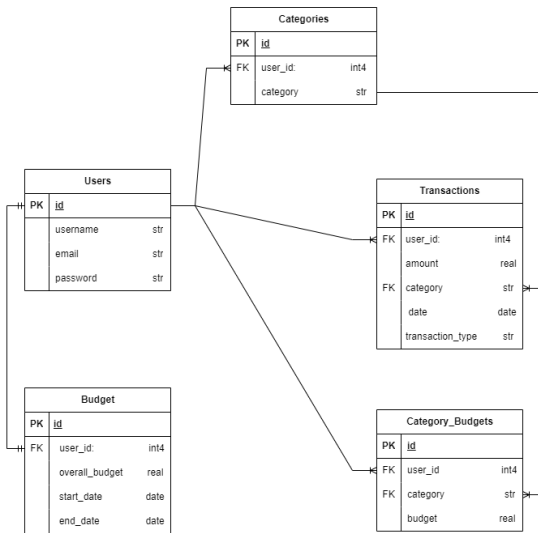
8

Діаграма класів



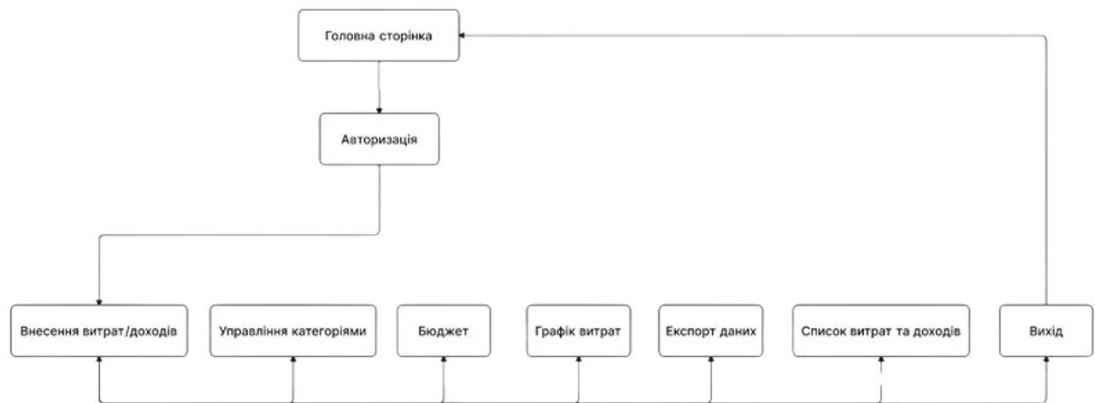
9

Схема баз даних



10

Мапа веб-додатку



11

ЕКРАННІ ФОРМИ



Головний екран додатку



Екран реєстрації



Екран авторизації

12

ЕКРАННІ ФОРМИ

ВАШЕ ФІНАНСОВЕ ЗДОРОВ'Я

Внесення витрат/доходів Управління категоріями Бюджет Графік витрат Експорт даних Список витрат та доходів Вихід

Внесення витрат/доходів

Вид транзакції:

Категорія:

Кількість:

Дата:

Додати

Екран внесення витрат та доходів

13

ЕКРАННІ ФОРМИ

ВАШЕ ФІНАНСОВЕ ЗДОРОВ'Я

Внесення витрат/доходів **Управління категоріями** Бюджет Графік витрат Експорт даних Список витрат та доходів Вихід

Управління категоріями

Нова категорія:

Додати категорію

- Продукти харчування
- Комунальні послуги
- Розваги

Екран управління категоріями

14

ЕКРАННІ ФОРМИ

ВАШЕ ФІНАНСОВЕ ЗДОРОВ'Я

Внесення витрат/доходів |
 Управління категоріями |
 Бюджет |
 Графік витрат |
 Експорт даних |
 Список витрат та доходів |
 Вихід

Бюджет

Загальний бюджет:

Дата початку:
дд.мм.гггг

Дата завершення:
дд.мм.гггг

Продукти харчування (Бюджет):

Комунальні послуги (Бюджет):

Розваги (Бюджет):

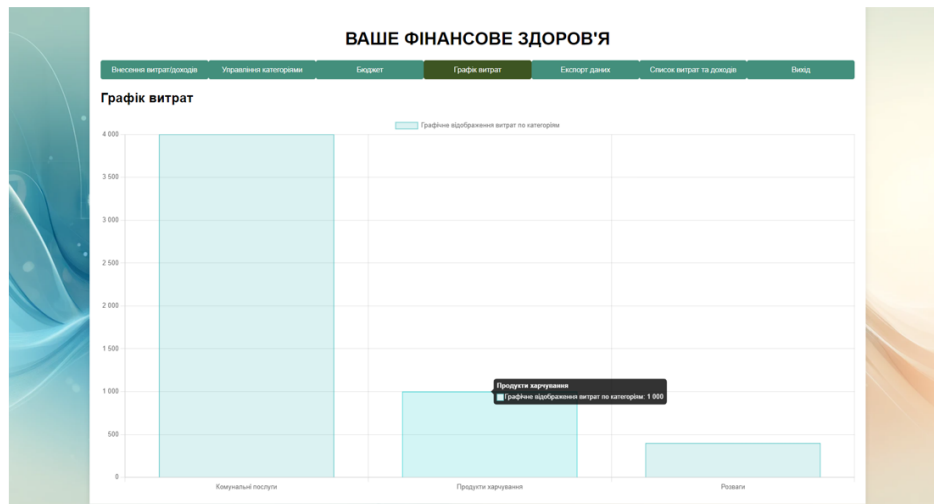
Створення бюджету

Загальний бюджет, грн. 24000
 Витрати, грн. 5400
 Термін дії бюджету: 2024-05-01 до 2024-05-31
 Продукти харчування Бюджет, грн. 5000, Витрати, грн. 1000
 Комунальні послуги Бюджет, грн. 4000, Витрати, грн. 4000
 Розваги Бюджет, грн. 2000, Витрати, грн. 400

Екран бюджету

15

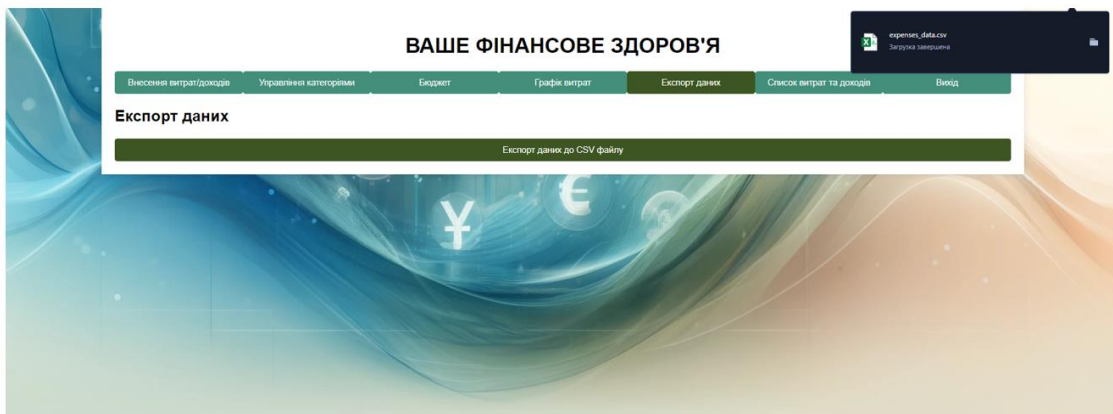
ЕКРАННІ ФОРМИ



Екран графіку витрат

16

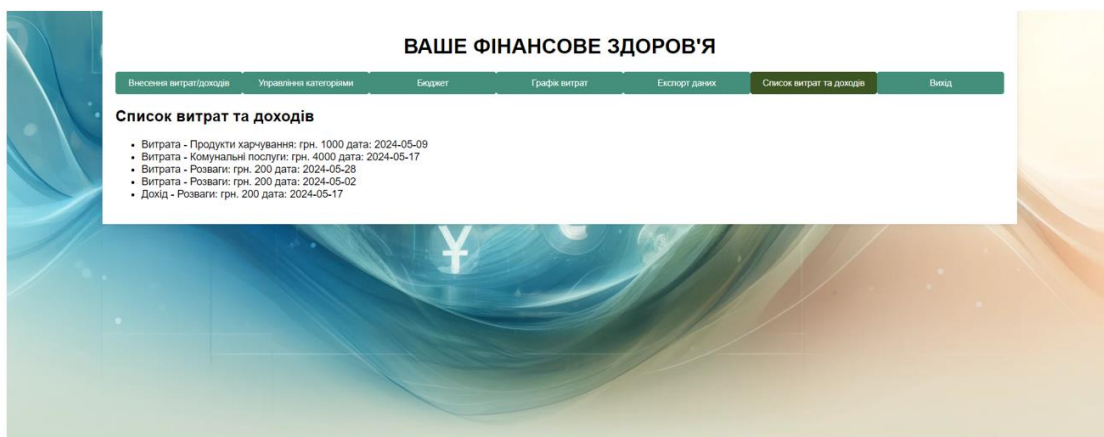
ЕКРАННІ ФОРМИ



Екран експорту даних

17

ЕКРАННІ ФОРМИ



Екран експорту даних

18

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Казьмірчук О.Г. Засоби мови програмування Python для захисту інформації //XI Всеукраїнська науково-практична конференція молодих учених «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ – 2024» (IT-2024). Збірник тез. 17.05.2024, Київський столичний університет імені Бориса Грінченка, Київ. К.: Київський столичний університет імені Бориса Грінченка, 2024, С. 238-240.
2. Садовенко В.С., Казьмірчук О.Г. Розробка програмного забезпечення аналізу витрат в домогосподарствах мовою Python //Всеукраїнська науково-технічна конференція “Застосування програмного забезпечення в інфокомунікаційних технологіях”. Збірник тез. 24.05.2024, ДУІКТ, Київ. К.: ДУІКТ, 2024, С. 88-90.
3. Садовенко В.С., Казьмірчук О.Г. Методи інтелектуального аналізу даних та засоби їх реалізації мовою Python у фінансовому плануванні домогосподарств //IV Всеукраїнська науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті». Збірник тез. 15.05.2024, ДУІКТ, Київ. К.: ДУІКТ, 2024, С. 56-57.

19

ВИСНОВКИ

1. Проаналізовано процес управління та аналізу витрат в домогосподарствах.
2. Проаналізовані засоби, які використовуються для ручного та автоматизованого управління та аналізу витрат в домогосподарствах.
3. Визначені переваги та недоліки існуючих програмних засобів, їх аналіз був використаний при розробці застосунку для управління та аналізу витрат в домогосподарствах. Розглянуті такі засоби як RocketGuard, Personal Capital та YNAB.
4. Розроблено функціональні та нефункціональні вимоги до застосунку для автоматизованого управління та аналізу витрат.
5. Спроектовано архітектуру, визначено класи для створення застосунку для автоматизованого управління та аналізу витрат в домогосподарствах, побудовані: діаграма варіантів використання, діаграма діяльності, діаграма класів.
6. Розроблено програмне забезпечення для автоматизованого управління та аналізу витрат в домогосподарствах з використанням мов Python, JavaScript; фреймворка Flask; бібліотек: Pandas, SQLAlchemy; СУБД SQLite; мови розмітки HTML; середовищ розробки PyCharm, Visual Studio Code; мови стилю CSS.
7. Проведено ручне тестування застосунку для автоматизованого управління та аналізу витрат у домогосподарствах.

20

ДЯКУЮ ЗА УВАГУ!